REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
03/20/2009	Final Technical Report	From 12/15/2005 to 12/20/2008

4. TITLE AND SUBTITLE

Development of a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM Phase III-B)

5a. CONTRACT NUMBER

N00014-04-1-0751

5b. GRANT NUMBER

N00014-04-1-0751, A00003

5c. PROGRAM ELEMENT NUMBER

09PR03236-CQ

6. AUTHOR(S)

Giacomo Marani, Junku Yuh, Song K. Choi, Son-Cheol Yu, Luca Gambella, Tae Won Kim, Allison Lyon, Kaikala Rosa, Aaron Hanai, Christopher McLeod, Edgar Gongora, Scott Weatherwax, Patrick Simmons, Greg Tamasahi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

University of Hawaii at Manoa
Department of Mechanical Engineering
2540 Dole St., Holmes 302, Honolulu, Hawaii 96822

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

ONR Code 33
Office of Naval Research
875 North Randolph Street
Arlington, VA 22203-1995

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

SAUVIM (Semi Autonomous Underwater Vehicle for Intervention Missions) involves the design and fabrication of an underwater vehicle that it is capable of autonomous interventions on the subsea installations, a task usually carried out by ROVs or human divers.

15. SUBJECT TERMS

Intervention AUV, Autonomous Manipulation, Underwater Autonomous Intervention

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE		149	Giacomo Marani
					19b. TELEPHONE NUMBER *(Include area code)* 808-956-2863

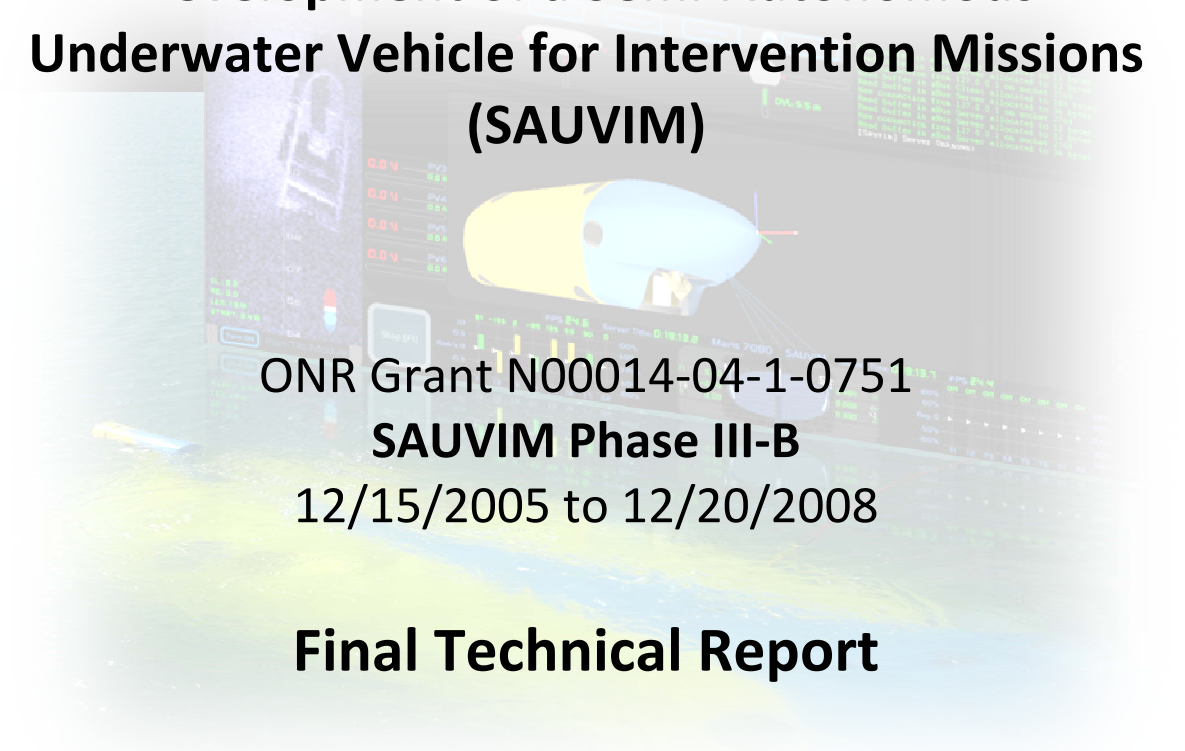
Reset

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

The
Autonomous Systems Laboratory
University of Hawaii at Manoa

2540 Dole Street, Holmes 302
Dept. of Mechanical Engineering
University of Hawaii
Honolulu, Hawaii 96822, USA
www.eng.hawaii.edu/~asl

Development of a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM)



ONR Grant N00014-04-1-0751

SAUVIM Phase III-B

12/15/2005 to 12/20/2008

Final Technical Report

March 20, 2009

Summary

Technical Development Report: December 15, 2005 – December 20, 2008

The SAUVIM proposal was submitted under the ONR Annual Announcement of the July 11, 1996 Commerce Business Daily, and the project officially began on August 1, 1997 with an 18-month, $2.237 million research fund from the Office of Naval Research's Undersea Weapons Technology Program directed by Mr. James Fein. The project was later extended at no cost till October 31, 2000.

Phase I of the SAUVIM project (ONR GRANT N00014-97-1-0961) officially began on August 1, 1997 with $2.237 million from the Office of Naval Research's Undersea Weapons Technology Program directed by Mr. James Fein. Additional funding of $1,445,000 for *Phase II-A* (the first part of Phase II) was received on May 1, 2000 (ONR GRANT N00014-00-1-0629). The second part of Phase II (*Phase II-B*) fund of $817,000 was received on June 17, 2002 (ONR GRANT N00014-02-1-0840). The third part of Phase II (*Phase II-C*) fund of $630,000 was received on August 1, 2003 (ONR GRANT N00014-03-1-0969). Phase III (*Phase III-A*) fund of $480,000 was received on October 1, 2004 (ONR GRANT N00014-04-1-0751, A0001). The second part of Phase III (*Phase III-B*) fund of $529,950 was received on December 15, 2005 (ONR GRANT N00014-04-1-0751, A0002). The Phase III-B has been extended at no cost until December 20, 2008. Table 1 summarizes the timeline and amounts of the SAUVIM grants until December 20, 2008.

In 1999, with the departure of Mr. James Fein from ONR, Mr. Chris Hillenbrand became the ONR Program Officer for the SAUVIM project. In 2002, Dr. David Drumheller became the new ONR Program Officer for the SAUVIM project. The Advisory Committee (AdCom) was formed to provide technical advice and direction by reviewing research directions and progress, and to provide advice and assistance in exploring potential applications and users. The six-member AdCom consists of Mr. Fred Cancilliere of Aquidneck Management Associates, Ltd (the former program director of the Naval Undersea Warfare Center), Dr. Alexander Malahoff of the University of Hawaii, Dr. Homayoun Seraji of the Jet Propulsion Laboratory, Dr. Paul Yuen of the University of Hawaii, and Mr. James Fein (the former ONR Program Officer for SAUVIM) of Carderock Division, Naval Sea Systems Command. Mr. Dick Turlington of the Pacific Missile Range Facility has retired and will be replaced by Mr. Clifton Ching.

The first progress report was submitted to ONR during Mr. Fein's site visit on October 28, 1997. The second progress report was submitted to ONR during the AdCom's site visit on February 24-25, 1998. The First Annual Report covering 1997-1998 was submitted to ONR in August 1998 and presented during the site visit on September 15-16, 1998. The fourth progress report was submitted during Mr. Hillenbrand's site visit on April 8, 1999. The Second Annual Report describing the overall technical progress of the project during the 1998-1999 year was submitted in July 1999. The next two ONR and AdCom site visits were on May 11, 2000 and November 14, 2000. A Final Report for Phase I was submitted to ONR in October 2000. During the next four ONR and AdCom site visits on October 29, 2001, July 18, 2002, February 18-19, 2003, and October 6, 2003, initial balancing and motion wet-tests of

SAUVIM were conducted, including various surge, heave, sway and yaw motions in the ROV mode for safety precautions. On May 27-28, 2004, SAUVIM performed underwater manipulation tasks and simple navigation motions in the pier area. These initial development results were publicly shown on October 21, 2004 for 80+ attendees of the Undersea Defense Technology (UDT) conference. A Phase II-B Final Report was submitted on March 31, 2005.
The next site visit was on July 14-15, 2005 where MARIS underwater manipulator demonstrated visual tracking performance in the water with a chess board image for target. Successively, during the review of April 27 and 28, 2006, various tasks were performed, including sea floor mapping, vehicle landing on an underwater platform, autonomous object recognition with a camera on the robot manipulator, and autonomous manipulation for target retrieval.

Start	End	Amount	Grant No.	Phase
08/01/1997	10/31/2000	$ 2,237,000.00	N00014-97-1-0961	SAUVIM Phase I
05/01/2000	12/31/2002	$ 1,444,993.83	N00014-00-1-0629	SAUVIM Phase II-A
06/17/2002	06/30/2004	$ 817,000.00	N00014-02-1-0840	SAUVIM Phase II-B
08/11/2003	06/30/2006	$ 630,000.00	N00014-03-1-0969	SAUVIM Phase II-C
10/01/2004	06/30/2006	$ 480,000.00	N00014-04-1-0751	SAUVIM Phase III-A
12/15/2005	12/20/2008	$ 529,950.00	N00014-04-1-0751	SAUVIM Phase III-B
	TOTAL:	$6,138,943.83		

Table 1: SAUVIM Grants

The present final report covers the Phase III-B of SAUVIM. This phase of the project has seen several major upgrades of the vehicle, including a new power source for enhanced autonomy, a new wireless communication link, the introduction of an Inertial Navigation system and a totally re-designed navigation controller with 6DOF performances.
The last site visit of SAUVIM Phase III-B, done on May 22-23 2007, presented the whole set of the new SAUVIM upgrades to the AdCom members.

In general, during every site visit, each SAUVIM research group gave a presentation of their tasks, objectives, and status. All AdCom reports for each site visit were submitted to ONR directly following each site visit. During this phase 7 people have been working under the SAUVIM project in ASL, consisting of 1 faculty member, 3 full-time researchers, 2 undergraduate interns, and 1 administrative assistant.

Objective

Many underwater intervention tasks are today performed using manned submersibles or Remotely Operated Vehicles in tele-operation mode. Autonomous Underwater Vehicles are mostly employed in survey applications. In fact, the low bandwidth and significant time delay inherent in acoustic subsea communications represent a considerable obstacle to remotely operate a manipulation system, making it impossible for remote controllers to react to problems in a timely manner.

Nevertheless, vehicles with no physical link and with no human occupants permit intervention in dangerous areas, such as deep ocean, under ice, in missions to retrieve hazardous objects, or in classified areas. The key element in underwater intervention performed with autonomous vehicles is autonomous manipulation, which refers to the capability of a robot system that performs intervention tasks requiring physical contacts with unstructured environments without continuous human supervision.

This challenging technology milestone is our long-term objective, through the development of a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM): *an undersea robot that can intelligently work with arms rather than just swim*, significantly advancing the Navy's ability for undersea intervention missions.

Today only few AUVs are equipped with manipulators. SAUVIM, at its current state of the art, is one of the first underwater vehicles designed to perform autonomous manipulation tasks.
The SAUVIM technical approach to underwater intervention involved the development of a robust autonomous manipulation framework over several years of researches.

Our current results represent an important passage toward the development of a higher level of autonomy for intervention AUVs, providing a cost-effective engineering solution to many new underwater tasks and applications that the fly-by type submersibles have not been able to handle.

Program Implementation

During the Phase III-B, research for SAUVIM was carried out by continued coordination of three organizations: *Autonomous Systems Laboratory* (ASL) of the University of Hawaii (UH), *Marine Autonomous System Engineering*, Inc. (MASE), and *Naval Undersea Warfare Center*, Newport (NUWC).

Junku Yuh has been the PI of the SAUVIM project from the ASL organization for Phase I, Phase II and Phase III-A. He continued to serve as PI during the Phase III-B of SAUVIM, for ASL. Frederick Cancilliere was the PI of the SAUVIM project from the NUWC organization for Phase II-B. Due to Mr. Cancilliere's retirement, Paul Temple has served as PI for Phase II-C and Phase III-A for NUWC. He will continue to serve as the PI for NUWC for Phase III-B. Song K. Choi has been the PI of the SAUVIM project from the MASE organization for Phase II and Phase III-A. He will continue to serve as the PI for Phase III-B for MASE.

The ASL is the primary research organization for the SAUVIM project. ASL staff members have developed key technologies of the SAUVIM project such as vehicle control system with real-time operating system, underwater navigation algorithm, sensor handling algorithm, sensor fusion, robot manipulator control system, and underwater image processing system. The ASL at UH has been used to train highly capable engineers and scientist to contribute to the underwater technologies society from various industries.
While Junku Yuh has been the official PI for ASL during all the SAUVIM Phases, the development of the current Phase III-B has been coordinated by Giacomo Marani from ASL, starting from December 2006. During the same period, Giacomo Marani served also as SAUVIM Acting PI. Tae Won Kim was the Project Co-Coordinator for Phase II and has been the Co-PI from ASL and the Project Coordinator of the SAUVIM project for Phase III-A and the beginning of Phase III-B, until December 2006.

MASE is the spin-off company from the ASL, UH. The key MASE staff members are former members of ASL, who were involved in the design, analysis, fabrication and testing of SAUVIM in Phase I and Phase II-A. Song K. Choi served as the SAUVIM Program Coordinator during Phase I and as the Associate Director in Phase II-A. He has been the PI of the SAUVIM project from the MASE organization for Phase II and Phase III-A. MASE's contribution to the proposed research is essential as MASE staff members' profound research experiences and skills especially with SAUVIM as well as their private sector environment are crucial factors to complete this project with respect to research outcome in industrial standards and future technology transfer. MASE plans to continuously provide engineering service for maintenance, modifications, and field operations of SAUVIM. SAUVIM is ultimately owned by UH and will be used for UH and Navy tasks as priorities.

NUWC is the main Navy laboratory where Navy's key projects in unmanned underwater vehicles (UUVs) have been carried out. NUWC possesses a great abundance of research and operational experience of UUVs, especially with the fly-by type AUVs. Mr. Cancilliere has served as a member of the SAUVIM Advisory Committee and is very familiar with the research objectives and progress of SAUVIM. Mr. Cancilliere initiated the maintenance, safety, and testing documentation during Phase II-B. During Phase II-C and III-A, Paul Temple has continued to lead the NUWC work by utilizing UNWC personnel who are already familiar with the SAUVIM project.

While their joint involvements are at different levels in the program, integrated research efforts of all three organizations are essential for the successful completion of the SAUVIM project. ASL is focused on the theoretical investigation and software development; MASE is focused on the experimental testing, hardware development, and sensor and power investigations; and NUWC is focused on the experimental implementation of the proposed research tasks, sea trials, and documentation. From December 2006, the overall technical coordination between the project entities, particularly among the research institute (ASL) and engineering service (MASE), was managed by Giacomo Marani.

The SAUVIM revised organizational diagram is shown in Figure A and a simplified SAUVIM schedule is shown in Figure D.

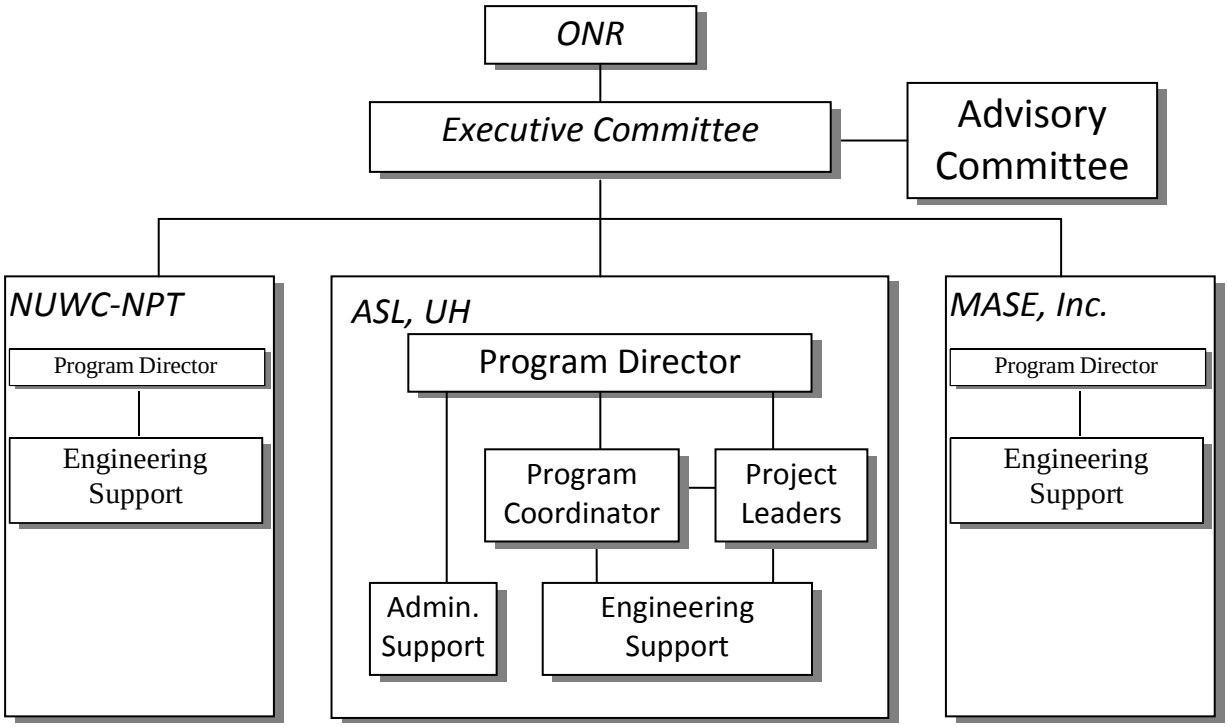


Figure A. SAUVIM Revised Organizational Diagram

Background

It is clear from various meetings with Navy experts and the autonomous underwater vehicle (AUV) community that there is a great need for improving undersea intervention capabilities in terms of autonomy, cost-effectiveness, and performance. Various underwater intervention tasks include underwater plug/unplug, construction and repair, cable streaming, mine hunting, and munitions retrieval. All underwater vehicles currently used for intervention missions are either manned submersibles or remotely operated vehicles with manipulators. These vehicle operations are expensive and often face a number of safety issues. Furthermore, their performance in terms of accuracy and efficiency are questionable, mainly due to human operator fatigue and the time delay in the man-machine control loop in an unstructured environment. Even though recent advances in sensors, communication, computers, and machine intelligence have made it possible to attempt to design advanced AUVs, the AUV development is still mostly directed toward a survey-oriented vehicles.

In literature there are only few examples of Intervention AUVs. These example include the OTTER I-AUV by the Stanford Aerospace Robotics Lab. OTTER, developed back in 1996, is a hover capable underwater vehicle which operates in a test tank at the Monterey Bay Aquarium Research Institute (MBARI). Current and past research includes texture-based vision processing for feedback control and real-time mosaicking, autonomous intervention missions, and hydrodynamic modeling of underwater manipulators. A study on automatic objects retrieval was done in [Wang95].

Another Intervention AUV, ALIVE, was developed in 2003 by Cybernetix. The aim of the EU-funded ALIVE project was to develop an Intervention-AUV capable of docking to a subsea structure which has not been specifically modified for AUV use. A description of the ALIVE vehicle was given in [Evans03].

The key element in underwater intervention performed with autonomous vehicles is autonomous manipulation. This is a challenging technology milestone, which refers to the capability of a robot system that performs intervention tasks requiring physical contacts with unstructured environments without continuous human supervision.

Intervention missions requiring physical contact with the surroundings in the unstructured, underwater environment always increase the level of risk in damaging the system and present completely different dynamic problems from fly-by, non-contact type operation. The overall motion of the vehicle-manipulator system is a high degrees-of-freedom (dof) operation due to additional dof of the manipulator added to the vehicle's six dof. Operation requires a high degree of precision and accuracy, which accomplishment if often complicated by the presence of an external disturbance such as a current. All of these issues present very complex engineering problems that have hindered the development of AUVs for intervention missions.

Autonomous manipulation systems, unlike teleoperated manipulation systems, that are controlled by human operators with the aid of visual and other sensory feedback, must be capable of assessing a situation, including self-calibration based on sensory information, and executing or revising a course of manipulating action without continuous human

intervention. It is sensible to consider the development of autonomous manipulation as a gradual passage from human teleoperated manipulation.
Within this passage, the most noticeable aspect is the increase of the level of information exchanged between the system and the human supervisor.

In teleoperation with ROVs, the user sends and receives low level information in order to directly set the position of the manipulator with the aid of a visual feedback. As the system becomes more autonomous, the user may provide only a few higher level decisional commands, interacting with the task description layer. The management of lower level functions (i.e. driving the motors to achieve a particular task) is left to the onboard system. The level of autonomy is related to the level of information needed by the system in performing the particular intervention. At the task execution level, the system must be capable of acting and reacting to the environment with the extensive use of sensor data processing.

The user may provide, instead of directly operating the manipulator, higher level commands during a particular mission, such as "unplug the connector". In this approach, the function of the operator is to decide, after an analysis of the data, which particular task the vehicle is ready to execute and successively to send the decision command. The low-level control commands are provided by a pre-programmed onboard subsystem, while the virtual reality model in the local zone uses only the few symbolic information received through the low bandwidth channel in order to reproduce the actual behavior of the system.

This report presents the solutions chosen to address the above issues for autonomous manipulation, developed during the course of the SAUVIM research project.

The proposed study is in response to current local and national needs for the development of this technology and will ultimately be useful in many intervention missions. SAUVIM technologies could be extended for *harbor security* that would be part of *homeland security*, one of our nation's current interests and concerns. One potential user is the Pacific Missile Ranging Facility (PMRF) of the U.S. Navy in the State of Hawaii.

Research

The SAUVIM project was proposed as a two-phase research and development program. Phase I had three parts: (1) to study the major research components, (2) to develop and integrate the basic software and hardware of SAUVIM, and (3) to test the vehicle in a shallow water environment. Phase II is a continuation and completion of the research and development of Phase I with water environment testing.

As stated in the original proposal, the project consists of five major components:

- Adaptive, Intelligent Motion Planning;
- Automatic Object Ranging and Dimensioning;
- Intelligent Coordinated Motion/Force Control;
- Predictive Virtual Environment; and
- SAUVIM Design.

During the Phase I period, there have been approximately sixty people supported by this ONR project. In 2007, there were 7 people working on the project in ASL - 1 faculty members, 3 full-time staff members, 2 undergraduate students, and 1 administrative assistant. The Advisory Committee was formed to provide technical advice and direction by reviewing research directions and progress, and to provide advice and assistance in exploring potential applications and users. The four-member Advisory Committee consisted of Mr. Fred Cancilliere of the Aquidneck Management Associates, Ltd., Dr. Alexander Malahoff of the University of Hawaii, Dr. Homayoun Seraji of the Jet Propulsion Laboratory, and Mr. Dick Turlington of the Pacific Missile Range Facility. Two additional members - Dr. Paul Yuen of the University of Hawaii and Mr. James Fein, the former ONR Program Director - have been included in the Advisory Committee.

- Adaptive, Intelligent Motion Planning (AIMP) – The AIMP aims at developing SAUVIM's motion planning, which is intelligent and adaptive in that the system is capable of decision-making at a task or mission level and can deal with unknown or time-varying environments. Motion planning for an AUV can be decomposed into path planning and trajectory generation, although they are not completely independent of each other. Path planning is a computation and optimization of a collision-free path in an environment with obstacles. Trajectory generation is the scheduling of movements for an AUV along the planned path over time. To simultaneously compensate for these objectives, a genetic algorithm (GA) based 3D-motion planner was studied both off-line and on-line cases. In general, and for any algorithms, an off-line case is when an environment is known and static, while an on-line case must be capable of modifications in response to dynamic, environmental changes. The utilization of GA-based approach has two advantages: 1) it is adaptive and 2) the dimension of space has less effect on performance than other methods.

  The AIMP software has gone through three version upgrades. The first was *Version 1.alpha*, which integrates the off-line and on-line algorithms in C with a graphic user interface using OpenGL. This software version was tested on the Autonomous Systems Laboratory's autonomous underwater vehicle - ODIN. The second was *Version 1.0*,

which integrates the path planning and trajectory generation algorithms. The third was *Version 1.1*, which optimizes the original software organization and data structures, and includes a database of mapping data on the main memory. Also, a Software Development Process (SDP) has been developed and implemented to oversee the various developments in software version changes. Several papers have been published in these subjects.

During an attempt to make an on-line version of AIMP in Phase II-C, it was found that there was no ending condition of genetic evolution to build a 3-D motion planner. There was no measure to guarantee optimality of the generated 3-D path or trajectory as well. Thus a conventional math-based motion planning method is implemented in Phase III-A, and a new motion planning algorithm has been investigated for complex motion in 3-D as well as minimizing computing burden.

Phase III-B has seen an increase in the degrees of freedom that the motion planner is able to handle, in order to better cope with the requirements of an underwater intervention. The conventional math-based motion has been extended in order to optimize the rotational and translational movements of the vehicle, allowing precise positioning of the robotic arm in the area of interest. This is usually different from the fly-by type AUV where the primary goal is to survey a generic area.

- Automatic Object Ranging and Dimensioning (AORD) – The main objective of the AORD is to develop a multiple sensor system to be utilized during SAUVIM's intervention missions to locate the target. This system originally consisted of three-sensors:
  1. Laser ranging sensor (LRS),
  2. Passive arm sensor (PA)
  3. Manipulator homing sensor (MHS)

  The laser ranger, the homing sensor, and the passive arm have all been designed and prototyped in the previous phases.

  The underwater prototypes of the LRS has been fabricated, assembled and tested, with the camera housings manufactured using 6061 aluminum with vacuum-sealed lens. The software has been developed using the prototypes.

  The PA, in its original configuration, was made of 6061-Aluminum, and it had two three-axis gimbaled joints and a single-axis hinge joint. The entire PA structure was compensated with mineral oil. It utilized the original software developed for the prototype. The kinematics of the PA has been re-verified using various symbolic math packages. The passive arm has also been rewired for optimal performance. It was simulated with the active arm to conduct feasibility studies in obtaining active manipulation position. However, after a long investigation, it was concluded that the PA cannot be easily deployed and retrieved in the water due to the lack of active power in the arm. Thus, an underwater version of the ultrasound motion tracker has been introduced as replacement of the passive arm system. Since there are no commercial versions of similar devices, a prototype version of the ultrasonic motion tracker had to be developed in house.

The original idea of the homing sensor was to use a dedicated PC104 computer with camera to detect a simple circular barcode. It was originally tested to confirm its performance in the water, and, despite results were good enough to use the bar code in the water, it suffered of obvious application limitations. During the past years, and especially starting with the Phase III-B, the idea shifted toward a more organized and range dependant Target Identification procedure.

The localization subsystem, that is the main support for the capabilities of the autonomous manipulation of SAUVIM, is performed by using and fusing different technologies (acoustical and optical) in order to guarantee a suitable, range dependent, level of reliability, precision and accuracy. The SAUVIM AUV switches through three main sensing methods in order to acquire reliable data. As shown in Figure B, the sensor technology changes according to the combination of range and accuracy needed.
In *long range* (over 25m), 375KHz image sonars are used for initial object searching. The accuracy in this range is necessary only to direct the vehicle toward the target zone.

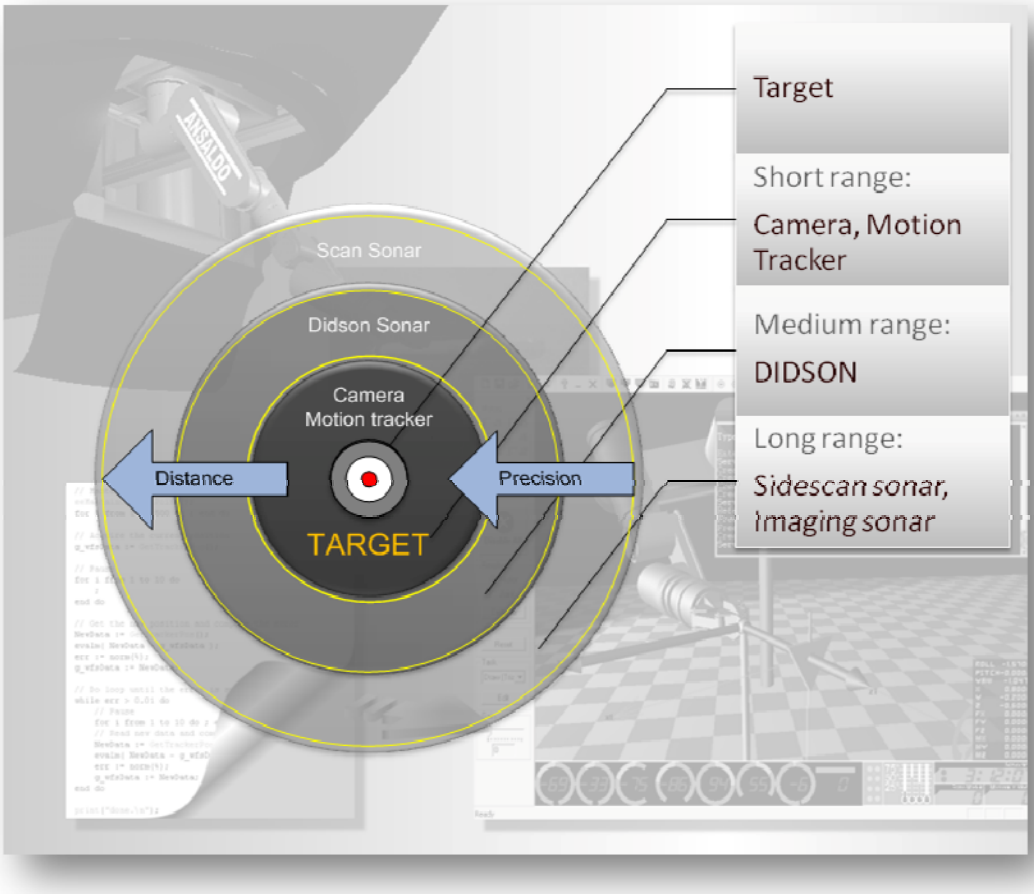


Figure B. The phases involved in a search for the target.

In *mid-range* (2-25m), a Dual frequency IDentification SONar (DIDSON) is used for object recognition and the vehicle positioning. This is the phase where the vehicle has to position itself in order to have the target confined within the manipulation workspace.

At this range, and in case of turbid water, it is virtually impossible to use conventional optical cameras to identify an object. This justify the use of the DIDSON, which has been used as a ranging sensor from Phase II-C onwards. During the Phase III-B our focus has been directed toward refining the algorithms of autonomous target identification with the DIDSON, thus allowing the SAUVIM vehicle to find a path toward the target area.

Finally, when the target is within the manipulator workspace, *short range* and high accuracy sensor are used in order to perform the actual intervention task. This goal is achieved with the combined use of underwater video cameras and the ultrasonic motion tracker described above, used to retrieve the real-time 6 DOF position of the target during the manipulation tasks. The device utilizes high frequency sound waves to track a target array of ultrasonic receivers. The use of 4 transmitters at the stationary positions with 4 receivers on the target can be used to determine the 6 DOF generalized position (rotation and translation) of the object.

- Intelligent Coordinated Motion/Force Control (ICM/FC) – The major objective of the ICM/FC is simple yet complex. The control of an AUV and its manipulator is a multi-bodied, dynamic problem of vast unknowns; therefore, this task was subdivided into four sub-tasks, which were:
  - Theoretical Modeling (TM)
  - Low-Level Control (LLC)
  - High-Level Control (HLC)
  - Dry Test Design and Set-up (DTDS).

  However, with the arrival of the 7-dof underwater manipulator, the TM and DTDS were combined to form a common group – Manipulator Control and Test Platform (MCTP). Also, a Localization and Navigation (LN) group was spun-off the LLC group due to the vastness and complexity of the LN material. The LN group was trying to devise a hybrid localization and navigation methodology that will suffice in understanding the geophysical, terrain-matching and dead-reckoning aspects for proper navigation. An integrated data fusion methodology was also being devised to quickly and correctly digest the immense amounts of data from the sensors, which undoubtedly has mass abundance of noise and errors. However, it was found that the map-based localization method is a task computationally intense and, despite this aspect, does not meet the accuracy requirement for the vehicle control. Thus, a Ultra Short Base Line (USBL) device has been used as a vehicle monitoring sensor as well as a position feedback sensor.

  The MCTP was developed to accelerate the progress in the **TM** and **DTDS** sub-tasks. With the acquirement of the MARIS 7080 manipulator and constraints in time, the focus has been changed to the development of the arm software in conjunction with the manipulator kinematics, dynamics, force-control and coordinated motion control modules. During the Phase II of SAUVIM the Maris 7080 manipulator initially ran off the VME bus system using VxWorks and Matlab with Simulink. Development in the "rapid prototyping, graphic software" has been the central point in enhancing the complex, underwater dynamic actions and reactions. The manipulator control code has

been developed to perform force/torque tasks, path optimization around singularity points, and collision avoidance techniques. Successively the development approach moved from the rapid prototyping mode to the stand-alone mode, in order to optimize the performances in the vehicle.

In phase II-B, a new parking procedure was developed and tested in the water. This is one of the most critical tasks of the manipulation system, due to the limited space for the manipulator in the vehicle. The Arm Programming Language (APL) was developed for high level control of the manipulator without changing system S/W, and an ultrasound motion tracker was interfaced (in the air) to the manipulator to get precise position feedback information, used either for calibration purposes and for initial dry tests of target tracking. The underwater version of the motion tracker is under development for substituting the passive arm which was originally planned to measure relative position/orientation between the target object and the vehicle. Preliminary results obtained during the Phase II-B showed a very high precision in position measurement.

In Phase III-A, image processing module in robot system was upgraded including new frame grabber and image processing library for Intel CPU. The phase III-B has seen further improvements of the camera system, with added procedure for auto-calibration to be performed directly on the target site (underwater) in order to compensate for the local water condition.

The **LLC** was created with two objectives: 1) to design and develop an advanced vehicle control system for navigation and hovering, and 2) to design and develop an advanced coordinate motion/force control system of the vehicle and manipulator during the intervention mode. However, with the creation of the LN group, the emphasis was on the integration of the localization and navigation techniques to the basic motion and hovering tasks. During the Phase III-A the development of the coordinated motion/force control system was being explored from two separate platforms. As the MCTP development continued, the LLC was optimizing the hovering and station-keeping methodologies on the ODIN vehicle. Various types of modern controllers, such as Adaptive controller, Disturbance Observer (DOB) controller, Adaptive DOB controller, and Neuro-fuzzy controller, were investigated in order to find the best controller for the underwater vehicle.

In all the SAUVIM Phases, the focus of the LN group has been on efforts in obtaining high performance in navigation and hovering. Before the current phase III-B the navigation and hovering techniques made use of the data from the on-board scan sonar, altimeter sonar, inertial navigation unit, Doppler Velocity Logger (DVL), and pressure sensors. The Ultra Short Base Line (USBL) and Global Positioning System (GPS) were added as a global vehicle position feedback sensor during underwater navigation and surface navigation, respectively.

However, at the end of the Phase III-A, it was evident how the accuracy and precision of this sensor system was insufficient, in particular conditions, during the manipulation tasks. Thus it was necessary, during the phase III-B, to introduce a more reliable Inertial

Navigation System aimed to produce the position data with the reliability necessary to the autonomous intervention.

This important change, together with a complete re-design of the navigation controller, allowed the SAUVIM vehicle to successfully perform in compliance with the precision, accuracy and stability requirements of our manipulation task.

Another important upgrade of the Phase III-B, aimed to improve the coordinate motion between the arm and the vehicle, was the standardization of all the communication protocols. This was accomplished with the extension of the xBus protocol, once dedicate to the arm subsystem only, to the entire vehicle. xBus showed a great flexibility in handling every kind of communication (data, program code, messages..) and thus it was chosen as the SAUVIM standard.

xBus uses a client-server approach for delivering information from and to each distributed module. Each subsystem (as a backset module or a generic sensor) embeds a custom TCP-IP client-server communication system (see [Marani05]). Within this architecture, every server can deliver the requested information on-demand to any number of clients, and this configuration allows a different utilization of the bandwidth, since every data is broadcasted only on demand.
This approach is similar to the Publish-Subscribe Middleware paradigm [Ben07], where the term `middleware` refers to the architecture software that coordinates the set of software modules collectively comprising the backseat-driver system running in the payload. Publish-subscribe middleware implements a community of modules communicating through a shared database process that accepts information voluntarily published by any other connected process and distributes particular information to any such process that subscribes for updates to such information.
In the SAUVIM approach the information is not published by a central database, but every source acts as a server that may send only the requested information to the requesting client. The distributed client-server architecture also provides a security hand-shaking mechanism, which provides direct feedback on the execution of any instance of data exchange. This is particularly desirable in issuing security commands (such as for aborting the mission).

HLC's objective is to develop a supervisory control module that will minimize human involvement in the control of the underwater vehicle and its manipulation tasks. In the gradual passage from human tele-operated manipulation to autonomous intervention, the most noticeable aspect is the increase of the level of information exchanged between the system and the human supervisor. In teleoperation with ROVs, the user sends and receives low level information in order to directly set the position of the manipulator with the aid of a visual feedback.

As the system becomes more autonomous, the user may provide only a few higher level decisional commands, such as "unplug the connector", interacting only with a higher level task-description layer. The management of lower level functions (i.e. driving the motors to achieve a particular task) is left to the onboard system. The level of autonomy

is related to the level of information needed by the system in performing the particular intervention.

With the above considerations in mind, the HLC module initially involved the development of high-level task planning where a mission is always composed of two parts: the goal and the method of accomplishment. In other words, "what do I need to do" and "how do I do it." Following this strategy, a new high-level architecture of vehicle control, named the Intelligent Task-Oriented Control Architecture (ITOCA), was developed for SAUVIM.

In phase III-B there was a major upgrade of this configuration. The high level control layer of both the manipulation and the navigation systems have been standardized and upgraded to a powerful custom programming language.

A software emulated CPU, where the mission control resides, hosts this new dedicated programming language developed in order to address the above issues [Marani05]. This language, suitable for real-time embedded control systems, offers at the same time flexibility, good performance, and simplicity in describing a generic complex task. Its layer abstraction approach allows an easy adaptation to the hardware-specific requirements of different platforms. For example, the same module can be found in the manipulator platform for describing a generic manipulation task and in the main navigation controller for driving the vehicle to the target area. The client-server approach allows the necessary communications between the arm and the navigation module.

The language is completely math-oriented and capable of symbolic manipulation of mathematical expressions. The last is an important distinctiveness from most of the currently available robot programming languages. The procedural approach has been chosen in order to enhance the performance while maintaining the flexibility required for executing complex tasks. It is particularly suitable for real-time embedded systems, where the interaction of a generic algorithm with the time is critical.

Predictive Virtual Environment (PVE) – The PVE is aimed at developing a supervisory monitoring system for SAUVIM to smoothly and realistically integrate mapping data with on-line sensory information even in the midst of delayed and limited information. This virtual reality (VR) based system must also be able to accurately predict the current status and location of the vehicle under these conditions. The development for the PVE has been modular. The various modules are: the SAUVIM Simulation Software (SSS); the SAUVIM Video Overlay Software (SVOS); the Communication Software (CS); and the artificial neural network (ANN) Video Prediction Software (VPS). In the Phases I and II of SAUVIM the SSS has been upgraded from its *Version 1* to *Version 1.1*, which includes the incorporation of a Magellan spaceball mouse, an accurate 3D graphical model of SAUVIM and the Maris 7080 manipulator, scene-smoothing methods using interpolation techniques, and an easy-to-use user interface. The SVOS was developed to overlay video images of the seafloor (texture and color) to the graphic images to provide a more accurate monitoring of the vehicle, manipulator and environment. The CS for

SAUVIM was an extension of the NSF's DVECS (Distributed Virtual Environment Collaborative Simulator) project. At that time, the DVECS system used a cellular phone to communicate the vehicle data from the test-site to the monitoring computer located on campus for data fusion. Experiments have been conducted with the ODIN AUV. The experiments of ODIN were projected via an ElectroHome Marquee 8500 CRT projector coupled with multiple Stereographics (SG) emitters and SG CrystalEyes glasses. Finally, the VPS has been tested, and, although in its early stage, with positive results.

Successively, due to the high maintenance costs of SGI workstations, the overall virtual reality and monitoring system, which includes the video prediction, has been transformed to a much more stable and inexpensive personal computing system, taking advantage of the emerging market of high performance hardware video accelerators (mostly targeted to PC games).

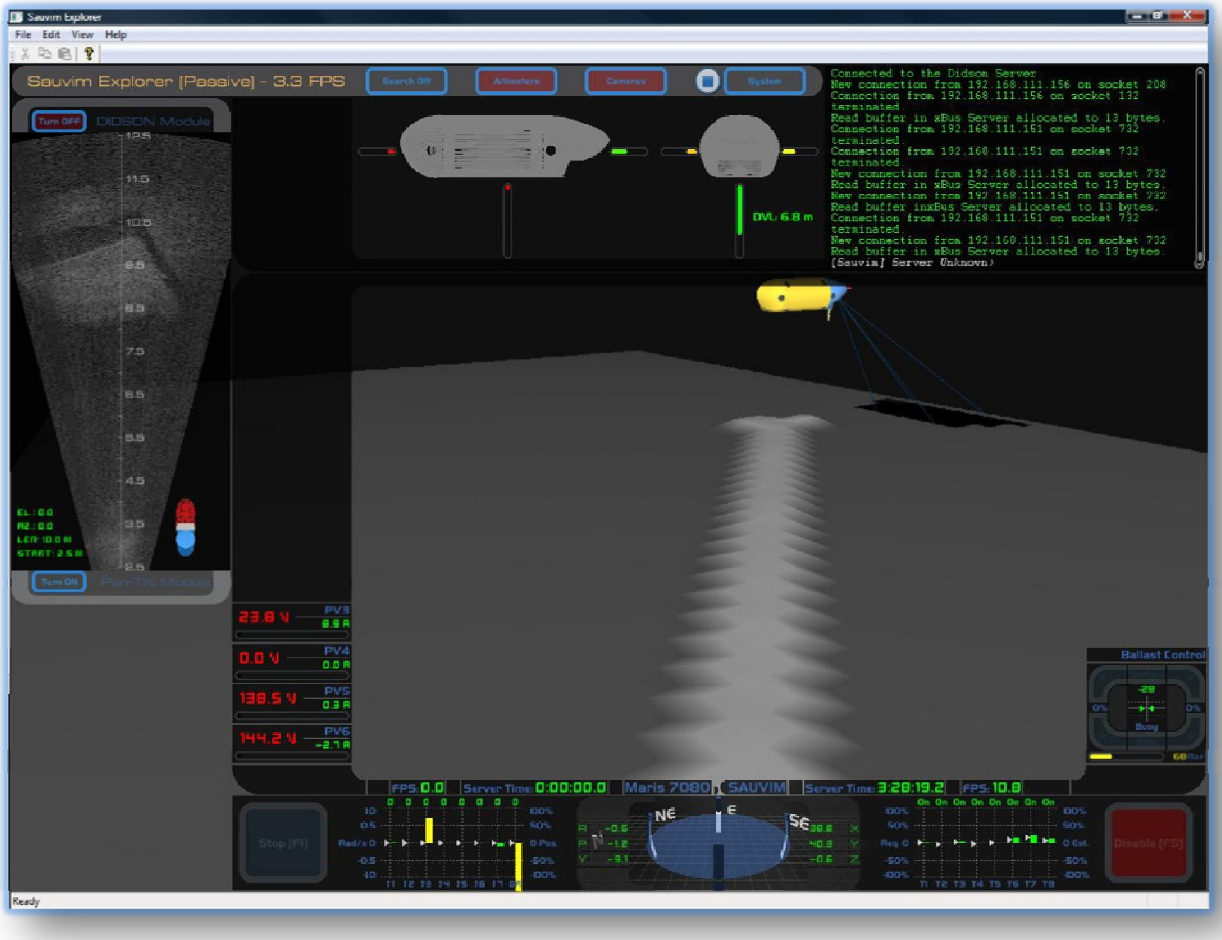


Figure C: Sauvim Explorer

MarisGL was, during the Phase II, the preliminary version of the virtual environment targeted to the MARIS 7080 Manipulator and making use of a standard OpenGL PC video accelerator. During the Phase III-A the application was extended in order to

introduce the vehicle model, mainly for collision avoidance verification. But the most important transition toward the global virtual environment happened in the current Phase III-B.

Here, the name of the application, once targeted to visualize only the configuration of the arm, has been changed to Sauvim Explorer (Figure C). Sauvim Explorer collects in a unified application the data from all the sensors of SAUVIM, including data from the DIDSON that can be overlaid over the graphical reconstruction of the floor.

It also hosts the remote console clients for both the Arm Programming Language and the Sauvim Programming Language servers, and may act as remote control (ROV mode) when a sufficient bandwidth channel is present. At this aim Sauvim Explorer contains software interface with several input device hardware, including 6 DOF space controllers.

This represents an enormous step forward toward the unification of the whole system, since it required a huge effort on the standardization of the communication protocol between every module of SAUVIM (sensors, actuators, controllers…). With this modular approach it is now extremely easy to add further sensor modules to SAUVIM and add their input and outputs to the SE application with a minimal effort.

SAUVIM Design (SD) – This task is still the main objective of the SAUVIM project. It is an effort to design and develop efficient, reliable hardware/software architectures of SAUVIM. Due to the immense demand of this task, it has been divided into five sub-tasks, which are:

- Reliable, Distributed Control (RDC)
- Mission Sensor Package (MSP)
- Hydrodynamic Drag Coefficient Analysis (HDCA)
- Mechanical Analysis and Fabrication (MAF)
- Mechanical-Electrical Design (MED).

The goal of **RDC** is to develop a reliable and efficient computing architecture for signal and algorithmic processing of the entire SAUVIM system. The proposed system is a multi-processor system based on a 6U VMEbus and the VxWorks real-time operating system. This system is capable of high processing throughput and fault tolerance. Currently the system consists of:

- Two VMEbuses, which are the navigation control system and the manipulator control system
- Two PC104+ computers dedicated to sensor data acquisition, processing and sharing;
- One PC104+ that hosts the video processing algorithms for the target detection and tracking system
- One PC104+ for the ultrasonic tracker (currently in development).

The main VMEbus, or the navigation control system, has one Motorola MC68060 CPU boards and a digital/analog I/O board, and two Pentium-M processor-based PC104+ boards, which share data through the Ethernet-based standard protocol xBus. The

navigation control system handles the communication, supervision, planning, low-level control, self-diagnostics, video imaging, etc.
The second VMEbus, or the manipulator control system, has one Motorola MC68060 CPU and several hardware-dedicated I/O board. One PC104 board aids the manipulator control system in performing the video processing operation necessary to detect and track the target. Data resulting from the video processing subsystem are shared with the whole SAUVIM system (including the Sauvim Explorer interface), again using the standard xBus protocol.
The manipulator control system, once independent and dedicated to the manipulator control, can now share its programming language subsystem with the navigation controller, a very important feature to perform underwater intervention.
Many of the hardware components have been tested and are interfaced with its respective software systems. Various optimization changes have been implemented to minimize communication and computation. The overall hardware and software architectures have been completed and integrated. Tests for the RTOS architecture has been integrated with the SAUVIM vehicle hardware and tested as individual components. The overall vehicle control with sensor feedback has been conducted at Snug harbor. This development will continue throughout the vehicle's development process.

The objective of the **MSP** is to provide semi-continuous records of underwater environment such as water depth, temperature, conductivity, computed salinity, dissolved oxygen, magnetic signature of the seafloor, pH, and turbidity, during the survey mode. In the intervention mode, the MSP also provides compositional parameters at a selected seafloor target, including pumped samples from submarine seeps or vents. The MSP is an independent system with its own PC 104 CPU and its own power supply residing in a separate pressure vessel. All of the sensors have been purchased and mounted, and an initial field test at the Loihi Seamount has been conducted. Other tests have been conducted to optimize the scientific sensor data-gathering capabilities. The communication from the MSP and the vehicle CPUs was initially based on RS-232C serial link.

The **HDCA** is used to determine the hydrodynamic coefficients via a numerical solution of full Navier-Stokes equations using PHOENICS, a commercial computational fluid dynamics (CFD) code. Initial results from the PHOENICS software have produced mixed results. The current vehicle fairing has produced a drag coefficient of 0.40; however, it has not yet been verified. Other CFD software and model testing is being conducted to verify the drag coefficient results before the implementation of the vehicle fairing on SAUVIM. There has been no significant development in this task group. The hydrodynamic coefficients will be obtained through vehicle motion experiments in the near future to aid in simulator developments.

The **MAF** has three objectives. Its primary goal is to design and fabricate composite pressure vessels with end caps and connector openings for full ocean depths taking stress, buckling, hydrothermal effects, and fatigue analysis into account; and its two secondary goals are to design and fabricate the SAUVIM fairing and to analyze the SAUVIM frame. A thorough analysis and comparison of the Ti-6Al-4V, AS4/Epoxy,

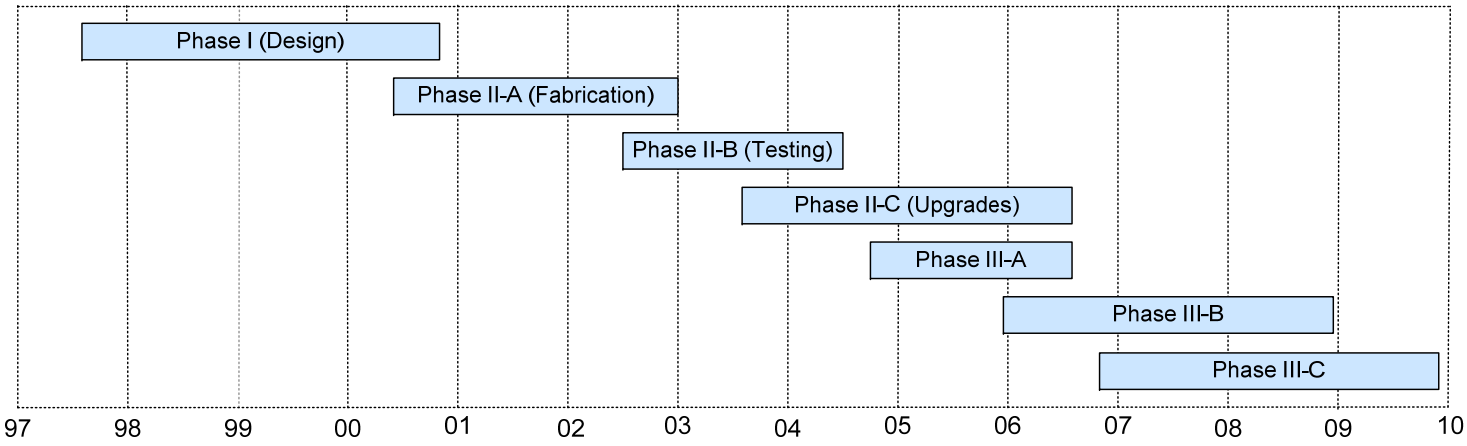
and AS4/PEEK pressure vessels manifest the advantage of composite materials in reduction of weight, size and strength. Using these results, a scaled model prototype using AS4/PEEK has been fabricated and tested. A 1/3 sized prototype is being fabricated and will also be tested. For the shallow water vehicle test, a full-sized, fiberglass pressure vessel with aluminum end caps have been manufactured and tested. These vessels are being used to determine the final hardware layout. The aluminum frame has been designed and fabricated. A full-ocean depth pressure vessel of AS4/PEEK has been developed and tested. However, due to several unknowns regarding composite pressure vessels, the vehicle has been equipped with 1000 meter-depth aluminum pressure housing. These aluminum housings will be used for the shallow and mid water depth experiments. The fairing analysis has been developed and expanded. After exploring various manufacturing and molding methods, the initial fairing was fabricated in-house in Phase II-B.

The **MED** is the integration of the mechanical and electrical components for SAUVIM. First, the design specifications were established for the fairing, frame, instrument pressure vessels, buoyancy systems, mission sensor, passive arm and robotic manipulator tasks. Second, after scrutinizing review of SAUVIM's major components - i.e. sensors, actuators and infrastructure - in terms of power consumption, compatibility, weight distribution, buoyancy distribution, hydrodynamic effects and task effectiveness, all major components have been purchased. Technical drawings of the vehicle frame, fairing, and related sub-structures have been completed. Most of the mechanical and electrical components have been fabricated and integrated with the overall electrical layouts. There were two wet-tests in Phase II-A, several autonomous shallow water tests were conducted in Phase II-B, and, from Phase II-C onwards, several vehicle navigation and underwater manipulation works.

The main body of this report is devoted to the detailed descriptions of the major technical developments and achievements during the period of Phase III-B, from the end of 2005 to the end of 2008.

A detailed description of the work prior December 2005 was given in the previous SAUVIM final reports.

Giacomo Marani
SAUVIM Project
March 20, 2009



Phase I : Conceptualization, Design, component Testing
Phase II-A : Vehicle design, Component design, Fabrication, Initial testing, Navigation testing in shallow water
Phase II-B : Design modification, Navigation testing in shallow water
Phase II-C : Upgrades, Manipulator testing in shallow water
Phase III-A : Upgrades, Navigation + Manipulator testing in shallow water
Phase III-B : Upgrades toward increased autonomy, Navigation + Manipulator testing
Phase III-C : Upgrades toward increased autonomy, Autonomous Intervention test

Figure D: SAUVIM: Simplified Gantt Chart

Table of Contents

Adaptive, Intelligent Motion Planning (AIMP)

Project Leader(s): Giacomo Marani
Personnel: Giacomo Marani
Past Project Leader(s): Dr. Tae Won Kim, Dr. Kazuo Sugihara & Dr. Song K. Choi
Past Personnel: Mr. John Smith, Dr. Shenyan Zhen, Mr. Haidong Chang, Ms. Hongshi Chen, Mr. Xihua Xu, Mr. Dwayne Richardson, Mr. Sonny Kim. Mr. Jangwon Lee & Mr. Yongcan Zhang

Objectives

This sub-project aims at developing the motion planning system for SAUVIM. It is intelligent and adaptive in the sense that the system is capable of decision-making at a task or mission level and can deal with an unknown or time-varying environment.

There are three basic objectives.

- To develop an off-line 3D motion planning algorithm.
- To develop an on-line 3D motion planning algorithm.
- To develop an adaptive, intelligent motion planning system by integrating the off-line and the on-line planning algorithms.

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

Introduction

While the previous SAUVIM phases have seen several studies on the Adaptive, Intelligent Motion Planning, during the current III-B phase it has been necessary to upgrade the trajectory generator algorithm in order to include path generation for generalized position (rotation and translation).
The previous algorithm for generating a trajectory in the Cartesian space has been found insufficient in particular conditions during an intervention task.

Trajectory Generation

The path-planning program produces a path represented by a sequence generalized positions coordinates:

$$X_i = \begin{bmatrix} \rho_i \\ \theta_i \\ \varphi_i \\ x_i \\ y_i \\ z_i \end{bmatrix} \tag{1.1}$$

where the six elements of the vector are, respectively, the *roll*, *pitch* and *yaw* angles of orientation of the vehicle (in Euler notation) and the Cartesian coordinates x, y and z.

We can associate a transformation matrix T_i to each generalized position X_i:

$$T_i = \begin{bmatrix} & & & x_i \\ & R_i & & y_i \\ & & & z_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.2}$$

where:

$$R_i = \begin{bmatrix} \cos(\phi_i)\cos(\theta_i) & -\sin(\phi_i)\cos(\rho_i)+\cos(\phi_i)\sin(\theta_i)\sin(\rho_i) & \sin(\phi_i)\sin(\rho_i)+\cos(\phi_i)\sin(\theta_i)\cos(\rho_i) \\ \sin(\phi_i)\cos(\theta_i) & \cos(\phi_i)\cos(\rho_i)+\sin(\phi_i)\sin(\theta_i)\sin(\rho_i) & -\cos(\phi_i)\sin(\rho_i)+\sin(\phi_i)\sin(\theta_i)\cos(\rho_i) \\ -\sin(\theta_i) & \cos(\theta_i)\sin(\rho_i) & \cos(\theta_i)\cos(\rho_i) \end{bmatrix} \tag{1.3}$$

is the rotation matrix correspondent to the Euler angles.
Given an initial generalized position X_0 and a final position X_1, the goal is to find a smooth generalized trajectory $X(t)$ through which the vehicle, starting from X_0, reaches X_1.
This can be accomplished with the following considerations.

<u>*Rotation.*</u>
Let R_1 and R_2 be the rotation matrices associated with the initial and final position respectively. Then it is possible to find a rotation vector that brings R_1 over R_2 .This correspond to find the axis-angle representation of the matrix:

$$R_{01} = (R_0)^{-1} R_1 \tag{1.4}$$

The axis-angle representation of a rotation, also known as the *exponential coordinates* of a rotation, parameterizes a rotation by two values: a unit vector indicating the direction of a directed axis (straight line), and an angle describing the magnitude of the rotation about the axis. The rotation occurs in the sense prescribed by the right hand grip rule.

The axis-angle representation of $R_{01} = R_{01}(\psi, \mathbf{v})$ is given by the quantity:

$$\begin{cases} \psi = \arccos\left(\dfrac{\text{trace}(R_{01})-1}{2}\right) \\ \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \dfrac{1}{2\sin(\psi)} \begin{bmatrix} R(3,2)-R(2,3) \\ R(1,3)-R(3,1) \\ R(2,1)-R(1,2) \end{bmatrix} \end{cases} \tag{1.5}$$

We can now describe a trajectory using the exponential representation:

$$R_{01}(t) = e^{\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \vartheta(t)} \tag{1.6}$$

where $\vartheta(t)$ describe a smooth trajectory from 0 to ψ, with $\vartheta(0) = 0$ and $\vartheta(t_1) = \psi$. Thus the rotation matrix associated to $X(t)$ becomes:

$$R(t) = R_0 R_{01}(t) = R_0 e^{\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \vartheta(t)} \tag{1.7}$$

Note that $R(0) = R_0$ and $R(t_1) = R_1$, as requested from our initial problem.

Translation.

The translational part of the problem is similar to the approach previously presented.
Let P_1 and P_2 be the Cartesian positions associated with the initial and final position respectively. Similarly to the rotational case, we can describe the trajectory from P_1 and P_2 using a parametric representation:

$$P(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = P_0 + (P_1 - P_2)\, s(t) \tag{1.8}$$

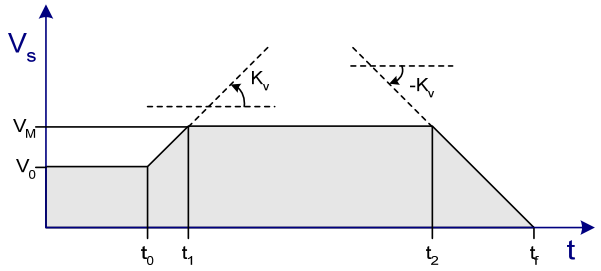
where $s(t)$ describe a smooth trajectory from 0 to 1, with $s(0) = 0$ and $s(t_1) = 1$.

Finally, combining the (1.7) and (1.8) together into the (1.2), we have:

$$T(t) = \begin{bmatrix} & R(t) & & P(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1.9}$$

The problem is now generating a smooth function for the two parameters $\vartheta(t)$ and $s(t)$. It can be accomplished with the following considerations.

Generation of trajectory.
The main constrain of our path is that must be continue up to its first derivative. We can thus first define a continuous function of the derivative to be trapezoidal.



Input data are x_0 , x_f , V_0 , V_m , K_v and t_0 . Then we can integrate the above profile obtaining:

$$x(t)=\begin{cases} x_0 & t=t_0 \\ x_0+\int_{t_0}^{t} V_0+K_v\cdot(t-t_0)dt & t_0\le t<t_1 \\ x_1+\int_{t_1}^{t} V_m dt & t_1\le t<t_2 \\ x_2+\int_{t_2}^{t} V_M-K_v(t-t_2)dt & t_2\le t<t_f \\ x_f & t\ge t_f \end{cases} \tag{1.10}$$

where:

$$x_1 = x_0 + \int_{t_0}^{t_1} V_0 + K_v\cdot(t-t_0)dt$$

$$x_2 = x_1 + \int_{t_1}^{t_2} V_m dt$$

$$x_f = x_2 + \int_{t_2}^{t_f} V_M - K_v(t-t_2)dt$$

Expanding above integrals we have:

$$x(t)=\begin{cases} x_0 & t=t_0 \\ x_0+\frac{1}{2}K_v\cdot\left(t^2-t_0^2\right)+V_0\cdot\left(t-t_0\right)-K_vt_0\cdot\left(t-t_0\right) & t_0\le t<t_1 \\ x_0+\frac{1}{2}K_v\cdot\left(t^2-t_0^2\right)+V_0\cdot\left(t_1-t_0\right)-K_vt_0\cdot\left(t_1-t_0\right)+V_m\cdot\left(t-t_1\right) & t_1\le t<t_2 \\ x_0+\frac{1}{2}K_v\cdot\left(t_1^2-t_0^2\right)+V_0\cdot\left(t_1-t_0\right)-K_vt_0\cdot\left(t_1-t_0\right)+V_m\cdot\left(t_2-t_1\right)- & \\ -\frac{1}{2}K_v\cdot\left(t^2-t_2^2\right)+V_m\cdot\left(t-t_2\right)+K_vt_2\cdot\left(t-t_2\right) & t_2\le t<t_f \\ x_f & t\ge t_f \end{cases} \tag{1.11}$$

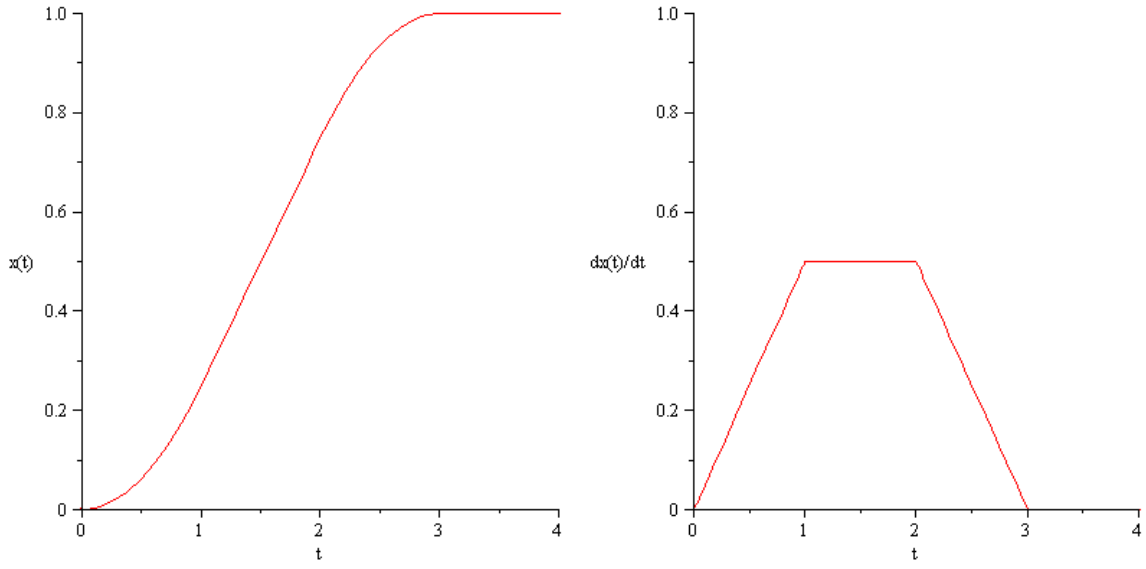
where:

$$t_1=t_0+\frac{V_m-V_o}{K_v} \tag{1.12}$$

$$t_2=t_f-\frac{V_m}{K_v} \tag{1.13}$$

$$t_f=\frac{2x_fK_v-2x_0K_v+2t_0K_vV_m+2V_m^2-2V_mV_0+V_0^2}{2K_vV_m} \tag{1.14}$$

An example of the function $x(t)$ for $V_0=0,\quad x_0=0,\quad t_0=0,\quad x_f=1,\quad K_v=0.5,\quad V_m=0.5$ is shown in the following figure.



It "smoothly" start from x_0 to reach x_f, with the constrain of the maximum velocity V_m and the maximum acceleration K_v. The time instant at which $x(t) = x_f$ is given by the (1.14). We can finally express our $\vartheta(t)$ and $s(t)$ as a function of $\bar{x}(t)$ in the following way:

$$s(t) = x(t) \tag{1.15}$$

$$\vartheta(t_1) = \psi \cdot x(t) \tag{1.16}$$

Substituting the (1.15) and (1.16) into the (1.9) we obtain the final form of the generalized trajectory of the vehicle.

Future Tasks (Phase III-C Tasks)

None

Automatic Object Ranging and Dimensioning (AORD)

Project Leader(s): Dr. Son-Cheol Yu & Dr. Giacomo Marani
Personnel: Dr. Son-Cheol Yu, Mr. Luca Gambella, Dr. Giacomo Marani, Dr. Tae Won Kim
Past Project Leader(s): Dr. Tae Won Kim, Dr. Junku Yuh, & Dr. Curtis S. Ikehara
Past Personnel: Mr. Marc Rosen, Mr. Mike Kobayakawa, Mr. Henrik Andreasson & Mr. Anders Andreasson, Mr. Aaron Hanai, & Mr. Oliver Easterday

Objectives

The main objective of the AORD is to develop a multiple sensor system to be utilized during SAUVIM's intervention missions to locate the target. The system will allow accurate vehicle positioning, workspace dimensioning and ranging, and manipulator homing to the task object. The localization subsystem, that is the main support for the capabilities of the autonomous manipulation of SAUVIM, is performed by using and fusing different technologies (acoustical and optical) in order to guarantee a suitable, range dependent, level of reliability, precision and accuracy

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

Overview

The original idea of the homing sensor was to use a dedicated PC104 computer with camera to detect a simple circular barcode. It was originally tested to confirm its performance in the water, and, despite results were good enough to use the bar code in the water, it suffered of obvious application limitations. During the past years, and especially starting with the Phase III-B, the idea shifted toward a more organized and range dependant Target Identification procedure.

The localization subsystem, that is the main support for the capabilities of the autonomous manipulation of SAUVIM, is performed by using and fusing different technologies (acoustical and optical) in order to guarantee a suitable, range dependent, level of reliability, precision and accuracy. The SAUVIM AUV switches through three main sensing methods in order to acquire reliable data. As shown in Figure B, the sensor technology changes according to the combination of range and accuracy needed.

In *long range* (over 25m), 375KHz image sonars are used for initial object searching. The accuracy in this range is necessary only to direct the vehicle toward the target zone.

In *mid-range* (2-25m), a Dual frequency IDentification SONar (DIDSON) is used for object recognition and the vehicle positioning. This is the phase where the vehicle has to position

itself in order to have the target confined within the manipulation workspace. At this range, and in case of turbid water, it is virtually impossible to use conventional optical cameras to identify an object. This justify the use of the DIDSON, which has been used as a ranging sensor from Phase II-C onwards. During the Phase III-B our focus has been directed toward refining the algorithms of autonomous target identification with the DIDSON, thus allowing the SAUVIM vehicle to find a path toward the target area.

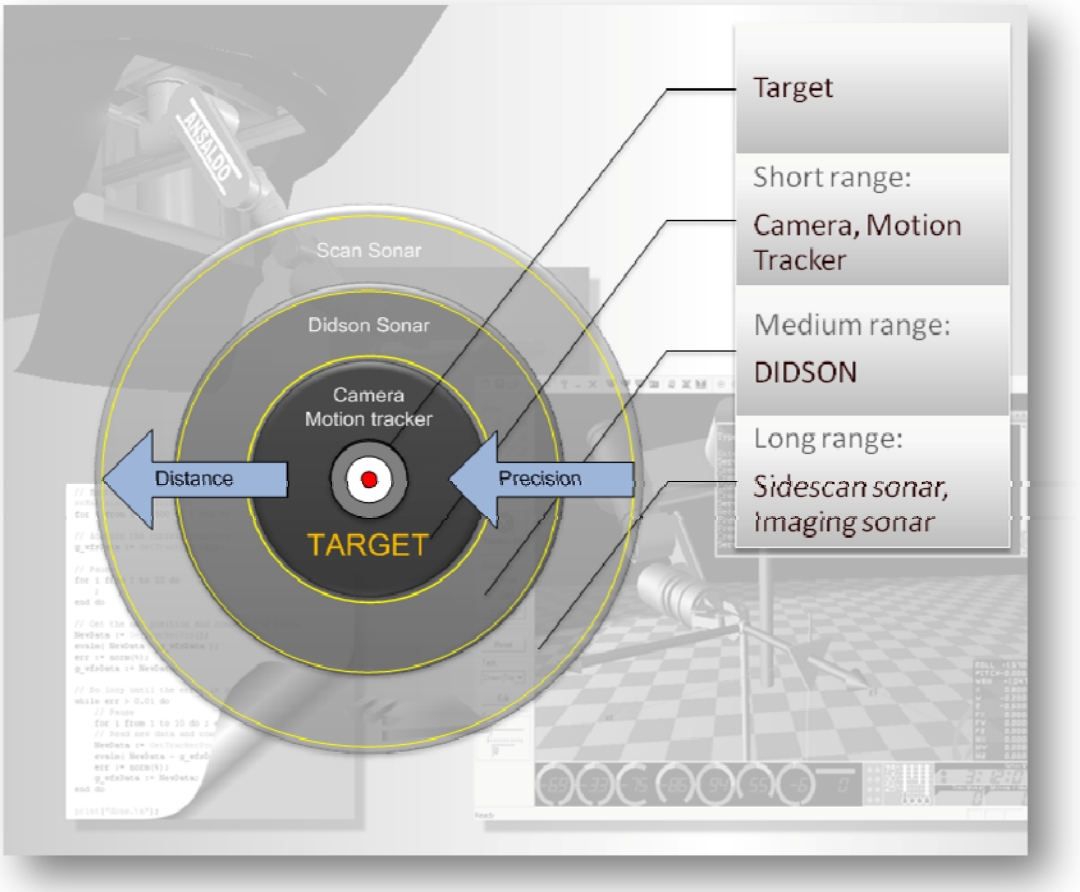


Figure AORD-1. The phases involved in a search for the target.

Finally, when the target is within the manipulator workspace, *short range* and high accuracy sensor are used in order to perform the actual intervention task. This goal is achieved with the combined use of underwater video cameras and the ultrasonic motion tracker described above, used to retrieve the real-time 6 DOF position of the target during the manipulation tasks. The device utilizes high frequency sound waves to track a target array of ultrasonic receivers. The use of 4 transmitters at the stationary positions with 4 receivers on the target can be used to determine the 6 DOF generalized position (rotation and translation) of the object.

Long-Range Object Recognition Method

Image scan sonar, Imagenex 881, is used to detect object in mid-range. Scanned acoustic image has been processed by conventional image processing technique. Without loss of generality, a cubic frame as shown in Figure AORD-2 is considered as a target for the initial test. The cubic frame(about 1m) was installed on Snug harbor seabed that is about 25 feet deep. In the image processing, the target was detected by the following.

Figure AORD-2. Target object (1m cubic)

In the firs, scan sonar data is converted as a 2 D black & white image as shown in Figure AORD-3. Then, the image morphology techniques, erosion and dilation, are used for removing island (small object or noise) and/or for filling crack due to noise. Using this filtered image, the labeling process starts. Labeling objects in the image and find the largest label, the seabed. After the labeling, finds the upper edge of the seabed, surface line.

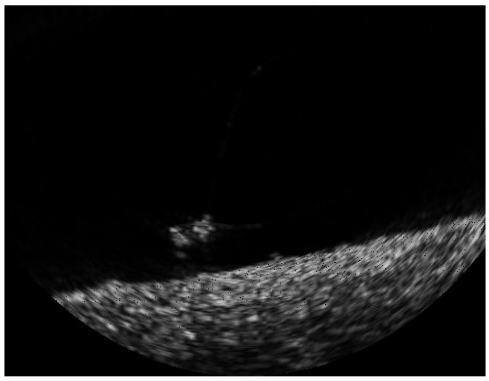


Figure AORD-3. Fig 5 Scanned raw image.

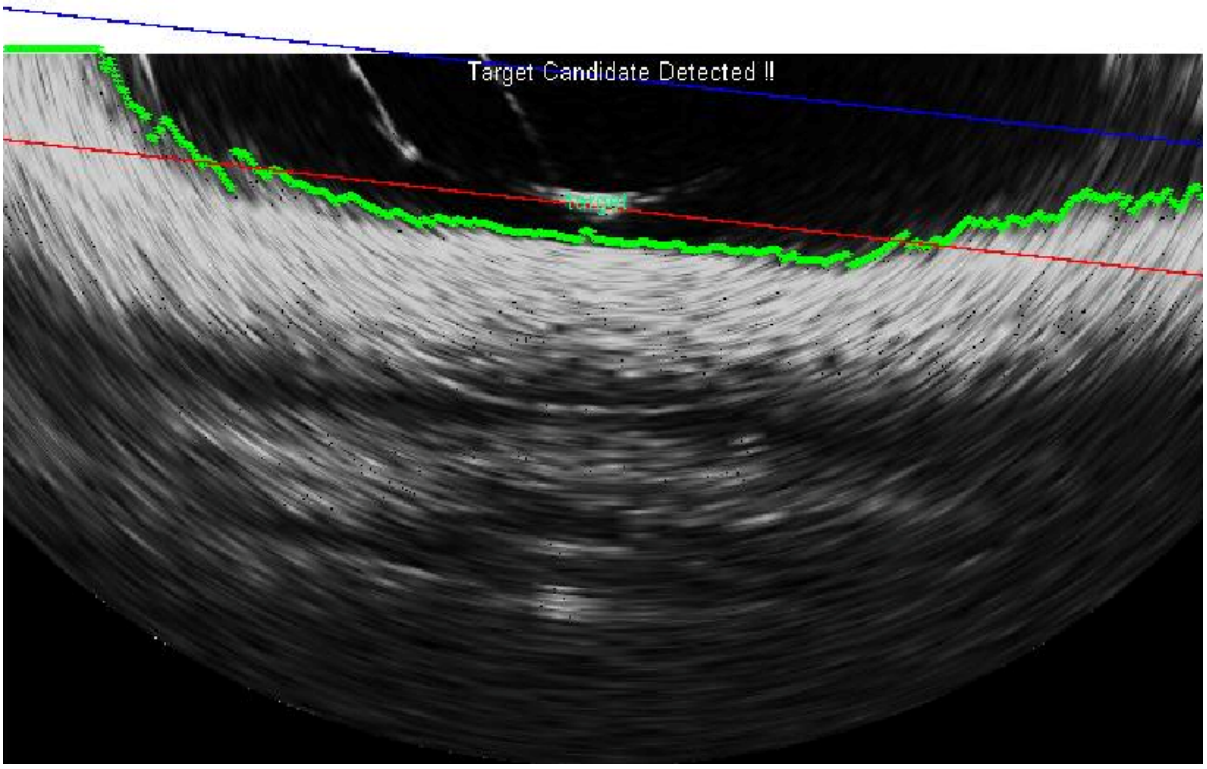


Figure AORD-4. Fig.6 Recognition in rolling condition

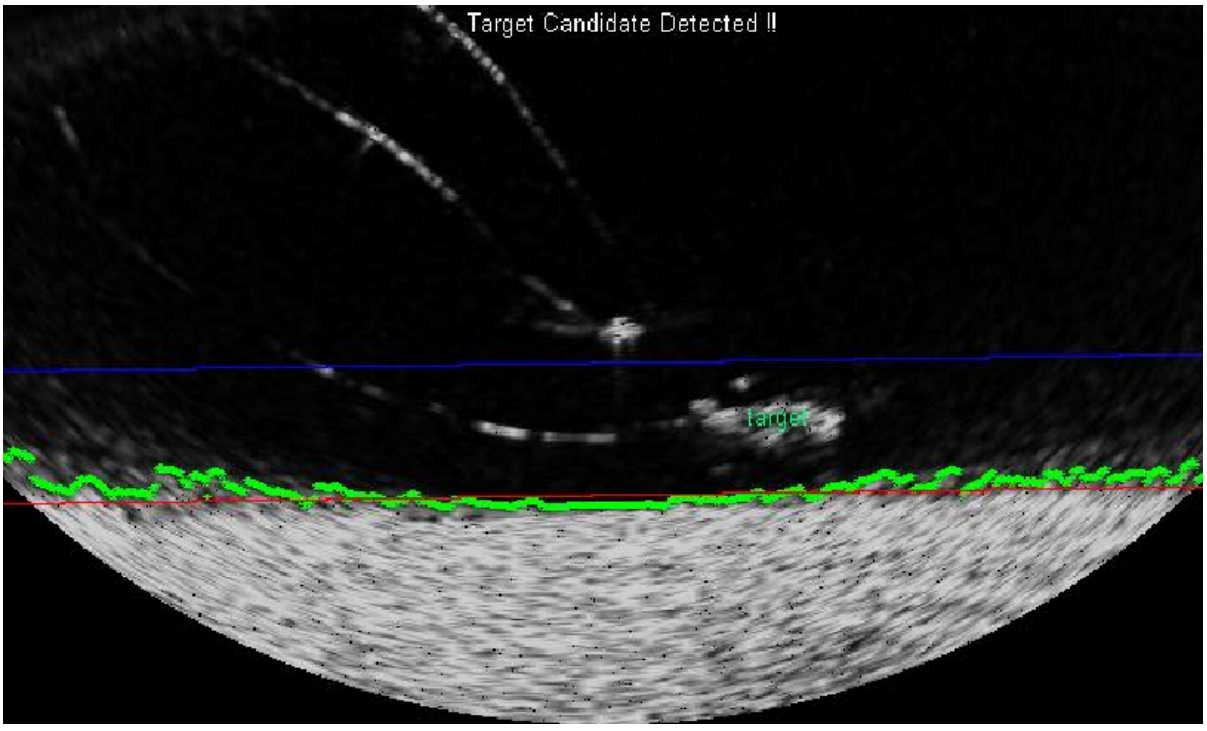


Figure AORD-5. Fig.7 Recognition with noise status

In general, Hough transformation is used for line detection in the image. Since actual seabed image hardly has long straight line, it is very difficult to detect the seabed level with simple Hough transformation. Thus least square method was implemented for detecting the seabed level. It gave better accuracy and robustness to detect the rough seabed level. In order to reduce the computational load, *a priori* information of target object is used. Size and height of the target are used for deciding target candidate and target searching area, respectively. The target candidate in the searing area is confirmed by its size and ratio. In this phase, various cases of scanned images have been tested to verify the performance of the proposed method as shown in Figure AORD-4 and Figure AORD-5. It was recognized correctly. Even the case of about +/- 20 degree rolling as shown in Figure AORD-4, the target was recognized exactly. In order to confirm the performance, the recognition algorithm was tested with various images such as background only (no target) images, noisy images, and object similar to the target, but different size. The algorithm shows good performance in any case. This recognition method will be improved to detect complicate

3D map building using 2D sonar:

Mid-range data, 2D sonar data can be re-use for localization of the vehicle and mapping. As shown in Figure AORD-6, the scanned 2D range data slices are able to build 3D map of seabed. Compared with side scan sonar or multi-beam sonar, this method had economic advantages and actively build 3D scene. Currently, the processing requires too heavy computing power to realize the real-time processing. We plan to optimize the process for the real-time and data feedback.

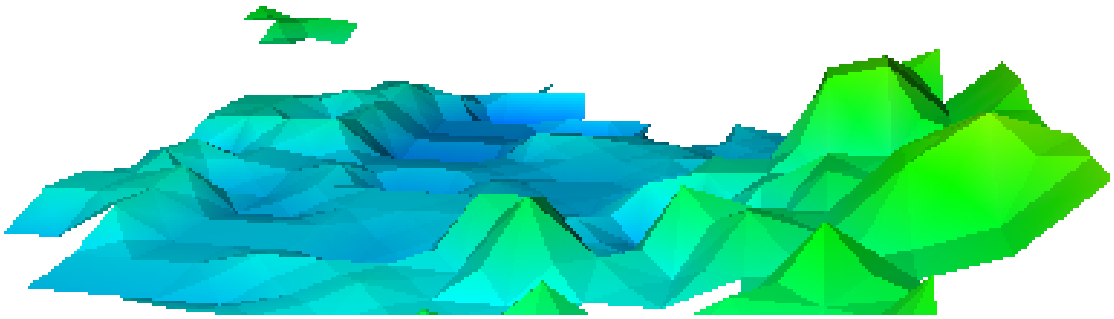


Figure AORD-6. Fig.8 3D map building using 2D data

Mid-range localization: the DIDSON sonar – Research Approach 1

To develop a robust underwater pattern recognition algorithm for Dual-frequency IDentification SONar (DIDSON) that was purchased during Phase II-B. The acoustic image recognition was planned and developed as the first step of AORD; a position/orientation-feedback system of SAUVIM with respect to the target object. However, since the

automatic underwater object detection in the turbid water is a big and hot issue, it is separated as a new technical branch as "Underwater Object Detection (UOD)."

The objectives of the UOD include the following;

- Noise reduction in underwater acoustic image
- Development of size invariant pattern recognition
- Increase robustness and reliability of recognition by using neural network
- Integration of pattern recognition algorithm in the SAUVIM system SW

1. Image preprocessing
   - Noise reduction with conventional noise filter
   - Data subsampling for reducing calculation burden as well as size invariant pattern recognition

2. Underwater acoustic image processing
   - Using Bidirectional Associative Memory (BAM) neural network for fast learning and retrieving data
   - Modify BAM for increasing learning capacity and robustness of classification
   - Test and upgrade of UOD with artificial targets in the water tank

Image processing for underwater acoustic image

Image processing has been a long time challenging problem for robots. Especially real-time image processing is hot issue for various unmanned vehicles systems such as Unmanned Aerial Vehicle (UAV), Unmanned Ground Vehicle (UGV), and Unmanned Underwater Vehicle(UUV). Since a vision system is the main environmental sensor for UAV and UGV, there have been so much of effort and good results for image processing. Unfortunately, however, UUV (or called Autonomous Underwater Vehicle, AUV) is in the different situation, because the use of the optical image processing is very limited only in the clean water and the acoustic image didn't have enough image resolution for applying image processing algorithm. Thanks to astonishing progress in technology these days, it is possible to get optical image-like precise acoustic image with high frequency sonar[1]. There have been some approaches to apply optical image processing methods to acoustic image[2, 5], but performance is still not good enough for automatic recognition. Among many image processing algorithms, a Bidirectional Associative Memory (BAM) is very famous neural network for binary image recognition because of its fast recognition speed from simple structure and good association characteristics of retrieving full image from partial image such as noisy or occluded image [3, 4, 6, 7].

Since the original BAM was designed to store image pairs in the correlation matrix, most of researches for the BAM were focused to increase storage capacity[9] and to guarantee recalling as it was trained [8]. In case of image classification/recognition, training image is provided as an input pattern. Thus an output pattern can be defined as a special binary code to show recognition result.

Bidirectional Associative Memory (BAM)

Associative memory is one of the major topics in neural networks. Kosko extended Hopfiled's one-layer unidirectional auto-associator neural network to two-layer and bidirectional associative memory [3, 4]. It can achieve hetero-association with a smaller correlation matrix as follows [6]:

There are N training pairs

$$\{ P_i = (A_i, B_i) \mid i = 1, \ldots, N \}, \tag{1}$$

where

$$A_i = (a_{i1}, a_{i2}, \ldots, a_{in}),$$
$$B_i = (b_{i1}, b_{i2}, \ldots, b_{in}).$$

And a_{ij}, and b_{ij} are either 0 or 1 in binary mode, or either -1 or 1 in bipolar mode.

Each pair is stored in associative memory by forming a correlation matrix, $X_i^T Y_i$, where X_i and Y_i are the bipolar mode of A_i and B_i, respectively. A number of associations can be stored by adding corresponding correlation matrices as

$$M = \sum_{i=1}^{N} X_i^T Y_i \ . \tag{2}$$

The BAM can retrieve one of the nearest pairs of trained data (A_i, B_i) from the network when any (α, β) pair is presented as an initial condition to the network. Starting with a value of (α, β), determine a finite sequence (α', β'), (α'', β''), ... until an equilibrium point (α_F, β_F) is reached, where

$$\beta' = \phi(\alpha M), \tag{3}$$
$$\alpha' = \phi(\beta' M^T), \tag{4}$$
$$\beta'' = \phi(\alpha' M), \tag{5}$$
$$\alpha'' = \phi(\beta'' M^T), \tag{6}$$
$$\ldots\ldots$$

and

$$\phi(F) = G = (g_1, g_2, \cdots, g_n,), \tag{7}$$
$$F = (f_1, f_2, \ldots, f_n), \tag{8}$$
$$g_i(f_i) = \begin{cases} 1 & , if \ \ f_i > 0 \\ \begin{cases} 0 \quad (binary) \\ -1 \quad (bipolar) \end{cases} & , if \ \ f_i < 0 \\ previous\ g_i & , if \ \ f_i = 0 \end{cases}. \tag{9}$$

Kosko [3] proved that this process will converge for any M. However, a pattern P_i can be recalled by the previous process if and only if this pattern is a local minimum of the energy surface[4, 6].

The overall structure of BAM is depicted in Fig. UOD-1.

Modified-BAM

As discussed in [4], if the network dimensionality increases, unintended fixed points (local minimums) called *spurious attractors* tend to increase due to simple correlation (Hebbian) encoding. The spurious attractors make the BAM malfunction; generate wrong recalling. Many researches have been performed to improve this problem, such as bipolar correlation encoding[4] and multiple training[6, 7]. However, they can't guarantee 100% recalling performance even with the trained data.

Example 1 : wrong recall

There is a good example of wrong recall from trained data in [6]. The BAM is trained with 3 pairs

A_1 = (100111000), B_1=(111000010)
A_2 = (011100111), B_2=(100000001)
A_3 = (101011011), B_3=(010100101).

Convert of these to bipolar form yields the (X_i, Y_i) namely

X_1 = (1 -1 -1 1 1 1 -1 -1 -1),
Y_1 = (1 1 1 -1 -1 -1 -1 1 -1),
X_2 = (-1 1 1 1 -1 -1 1 1 1),
Y_2 = (1 -1 -1 -1 -1 -1 -1 -1 1),
X_3 = (1 -1 1 -1 1 1 -1 1 1),
Y_3 = (-1 1 -1 1 -1 -1 1 -1 1).

The correlation matrix M is calculated as

$$
\begin{aligned}
M &= X_1^T Y_1 + X_2^T Y_2 + X_3^T Y_3 \\
&= \begin{bmatrix}
-1 & 3 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
3 & -3 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & -1 & -3 & 1 & -1 & -1 & 1 & -3 & 3 \\
3 & -1 & 1 & -3 & -1 & -1 & -3 & 1 & -1 \\
-1 & 3 & 2 & 2 & -1 & -1 & 1 & 1 & -1 \\
-1 & 3 & 1 & 1 & -1 & -1 & 1 & 1 & -1 \\
1 & -3 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & -1 & -3 & 1 & -1 & -1 & 1 & -3 & 3 \\
-1 & -1 & -3 & 1 & -1 & -1 & 1 & -3 & 3
\end{bmatrix}
\end{aligned}
\tag{10}
$$

In case that the BAM has an input exactly same as X_2, Kosko's decoding process is supposed to recall Y_2. However, actual recall is

$$
\begin{aligned}
X_2M &= (5\ -19\ -13\ \ -5\ \ 1\ 1\ -5\ -13\ 13) \\
\phi(X_2M) &= (1\ -1\ -1\ -1\ \ \underline{1\ \ 1}\ -1\ -1\ 1) \\
&\neq Y_2 = (1\ -1\ -1\ -1\ \underline{-1\ -1}\ -1\ -1\ 1).
\end{aligned}
$$

Even though the BAM has input with training data, A_2, it retrieved untrained data ($\neq B_2$) because the data pair (A_2, B_2) is not a local minimum. Multiple training was proposed by Wang *et. al*. [6] in order to recall all trained data. However, the number of training was decided by trial-and-error. Maximum training trials to guarantee recalling all data is calculated in [7].

It is known that the association performance of the BAM relies on ratio of 0 and 1 (or -1 and 1 in bipolar mode). If appearance of 0's and 1's in the training pair is almost same, overall recall performance is significantly increased. In this example, B_2 has two 1's and seven 0's. It is easy to expect that this unbalanced appearance makes some bad effect in the correlation performance.

Example 2 : equal distribution

In order to solve wrong recalling problem in Example 1, let's think about equal distribution of 1's and 0's as

$$
\begin{aligned}
B_1^* &= (1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 1), \\
B_2^* &= (1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1), \\
B_3^* &= (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0).
\end{aligned}
$$

Bipolar vector of B_i^* is calculated as

$$
\begin{aligned}
Y_1^* &= (\ 1\ \ 1\ \ 1\ -1\ -1\ -1\ \ 1\ -1\ \ 1), \\
Y_2^* &= (\ 1\ -1\ \ 1\ -1\ \ 1\ -1\ \ 1\ -1\ \ 1),
\end{aligned}
$$

$$Y_3^* = (\,-1\ \ 1\ -1\ \ 1\ -1\ \ 1\ -1\ \ 1\ -1).$$

Then, the matrix M^* is recalculated as

$$\begin{aligned} M^* &= X_1^T Y_1^* + X_2^T Y_2^* + X_3^T Y_3^* \\ &= \begin{bmatrix} -1 & 3 & -1 & 1 & -3 & 1 & -1 & 1 & -1 \\ 1 & -3 & 1 & -1 & 3 & -1 & 1 & -1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \\ 3 & -1 & 3 & -3 & 1 & -3 & 3 & -3 & 3 \\ -1 & 3 & -1 & 1 & -3 & 1 & -1 & 1 & -1 \\ -1 & 3 & -1 & 1 & -3 & 1 & -1 & 1 & -1 \\ 1 & -3 & 1 & -1 & 3 & -1 & 1 & -1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 \end{bmatrix} \end{aligned} \tag{11}$$

In order to confirm the performance of the new matrix M^*, three trained data are inputted to new matrix M^* as

$$\begin{aligned} X_1M^* &= (\,1\ \ 17\ \ 1\ -1\ -17\ -1\ \ 1\ \ -1\ \ 1\,) \\ \phi(X_1M^*) &= (\,1\ \ 1\ \ 1\ -1\ \ -1\ \ -1\ \ 1\ \ -1\ \ 1\,) = Y_1 \\ X_2M^* &= (5\ \ -19\ \ \ 5\ \ -5\ \ 19\ \ \ -5\ \ \ 5\ \ -5\ \ 13\,) \\ \phi(X_2M^*) &= (\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\,) = Y_2 \\ X_3M^* &= (\,-11\ \ 13\ -11\ \ 11\ -13\ \ 11\ -11\ \ 1\ \ -11\,) \\ \phi(X_3M^*) &= (\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1\ \ 1\ \ -1\,) = Y_3 \end{aligned}$$

It is remarked that even appearance of 0's and 1's (or -1's and 1's in bipolar mode) in the training pattern increase the overall neural network performance. It doesn't need any special techniques to recall proper trained data.

Output pattern for image recognition

The original purpose of the BAM is to store image pairs in the neural network. This network structure for bidirectional association is very useful to retrieve (recall) images from partial image information such as noisy or occluded image[3, 6].

In case of applying the BAM to image recognition or classification, only one image is given for each learning pattern. Thus, it is needed to define the other image which should be distinguished from other data. To do this, very simple data pattern is used as an output pattern of the training data in this paper. To be specific, in case of small number of training images, the output pattern can be determined as

$$
\begin{aligned}
&B_i = (t_1(i) \quad t_2(i) \quad \cdots \quad t_n(i)), \\
&t_j(i) = (t_{j1}(i) \quad t_{j2}(i) \quad \cdots \quad t_{js}(i)), \\
&t_{jk}(i) = \begin{cases} 0, & if \quad j \neq i \\ 1, & if \quad j = i \end{cases},
\end{aligned} \tag{12}
$$

where n is number of training images, s is size of dummy pack.

For example, if four images are considered to be trained, and size of dummy pack for each image is 3, then the output patterns are

$$
\begin{aligned}
B_1 &= (1\,1\,1 \quad 0\,0\,0 \quad 0\,0\,0 \quad 0\,0\,0), \\
B_2 &= (0\,0\,0 \quad 1\,1\,1 \quad 0\,0\,0 \quad 0\,0\,0), \\
B_3 &= (0\,0\,0 \quad 0\,0\,0 \quad 1\,1\,1 \quad 0\,0\,0), \\
B_4 &= (0\,0\,0 \quad 0\,0\,0 \quad 0\,0\,0 \quad 1\,1\,1).
\end{aligned}
$$

Since the Hebbian distance between data are all maximum, overall learning and recalling performance will be also maximum. In case of large number of training images, binary encoding method can be used in order to make Hebbian distance as large as possible.

It is remarked that even though output patterns are generated with maximum Hebbian distance, appearances of 0's and 1's are still not even. In order to make it even appearance, output patterns are newly defined by adding dummy images with complement data as follows;

$$
B_i^* = [\,[B_i]\,[\text{complement of } B_i]\,] \tag{13}
$$

So, B_1^* can be recalculated as

$$
B_1^* = (\underbrace{111\ 000\ 000\ 000}_{\text{Original } B_1}\ \underbrace{000\ 111\ 111\ 111}_{B_1 \text{ Complement}}).
$$

With complementary dummy data, overall network performance won't be degraded regardless of output patterns.

Optical image vs. Acoustic image

As described in previous researches [5, 6], the process of imaging in acoustic camera is different from that of optical camera. Despite the optical image shows the intensity of the reflected light from each point on the object, the acoustic image shows the intensity of reflected acoustic energy from the object, and the distance (specifically, traveling time of acoustic wave) from the sonar to the object. And, due to imaging mechanism and mechanical characteristics of the acoustic camera system, it is impossible to get pin-point distance of each point on the object. Thus, shadow of the object has much valuable information about object's shape. Detail imaging mechanism of the acoustic camera is described in [9].

There are so many researches to recognize or classify underwater objects in the water with sonar[8], but the results were not so significant. However, thanks to fast technology revolutions in computer engineering, high frequency precise sonar systems are introduced [1]. Even though acoustic system is different from optical system, high frequency sonar helps to get optical-like image as shown in Fig.UOD-2. With this acoustic camera, most of image processing algorithm for optical image can be applied to acoustic image.

BAM for image classification

In order to apply the BAM to underwater image classification, four target objects are used as shown in Fig. UOD-2. **D**ual-frequency **ID**entification **SON**ar (DIDSON) [1] is used to acquire acoustic images of objects in the water. Examples of real DIDSON image are shown in Fig. UOD-4. In order to apply the BAM to acoustic image, the raw shadow image is inversed, and binarized as shown in Fig. UOD-4.

The calculation speed of the BAM relies on the size of the BAM, because most of calculation in BAM is multiplication of a vector and a matrix. Without loss of generalization, the size of the input image is determined as 15×10. This size is good enough to identify objects as shown in Fig. UOD-5.

Compare to optical image, acoustic image has two types of information such as shadow and high light part, as shown in Fig. UOD-3. Since the shadow shows most of object's shape, it is used for image processing in this paper.

As described in Sec. 3, output patterns are needed for image processing with the BAM. Since the sample data is only four images, size of pack is selected as 10 in this paper, but the size is not limited. Thus, overall output pattern is composed of 80 binary data including complement dummy data of 4×10 = 40 binary data. After changing the input image matrix to vector form, M is calculated as large as 150×80 data.

For verification of recall performance and robustness against noise, various image taken from different distances and angles, and noised image are applied to the trained BAM. Test images are shown in Figs. UOD-6 and UOD-7.

Table UOD-1 shows test results with various sample images along with B/W noise from 30% to 70%. There are some failures in 70% noisy images of Tabbed Cylinder and Cone. However, this much of noise is also hard to recognize by human as shown in Fig. UOD-7(c). And, there is misclassification between Cylinder and Cube, because size normalized Cylinder and Cube images look almost same. However, overall classification with the modified BAM is very reliable and stable in case of less than 50% noise.

Concluding Remarks

The BAM is modified for acoustic image processing by generating output patterns with complementary dummy data. This modification makes the BAM more reliable against

noise. Especially, it drives overall energy level to the local minimum so that it helps to recall trained data from distorted image, noisy image, and occluded image. The test results show robustness of the modified BAM for acoustic images.

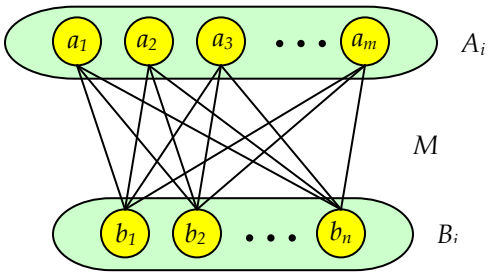
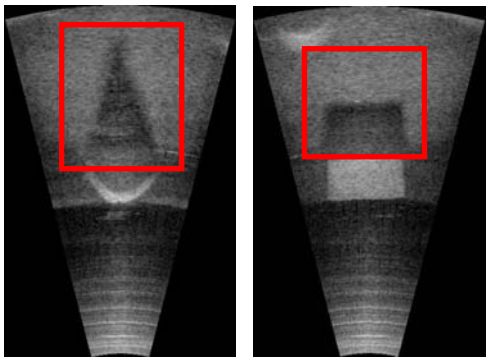


Fig. 1. Structure of the BAM

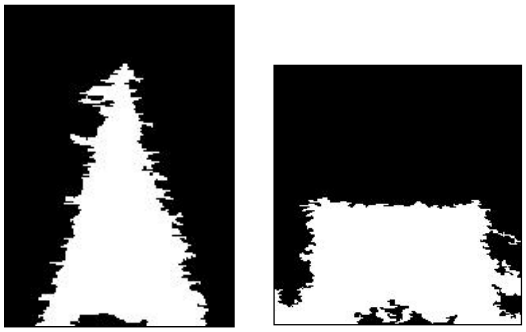
(a) Tapped cone (b) Cone (c) Cube (d) Cylinder

Fig. UOD-2. Example of underwater target objects



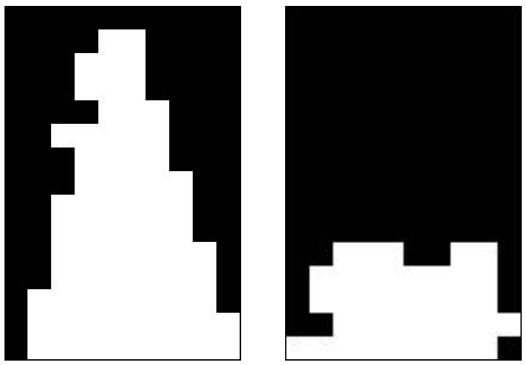
(a) Cone (b) Cube

Fig. UOD-3. DIDSON images of underwater objects



(a) Cone (b) Cube

Fig. UOD-4. Examples of binary image of underwater object taken by DIDSON



(a) Cone (b) Cube

Fig. UOD-5. Example of BAM input images (15×10) from Fig. UOD-3.

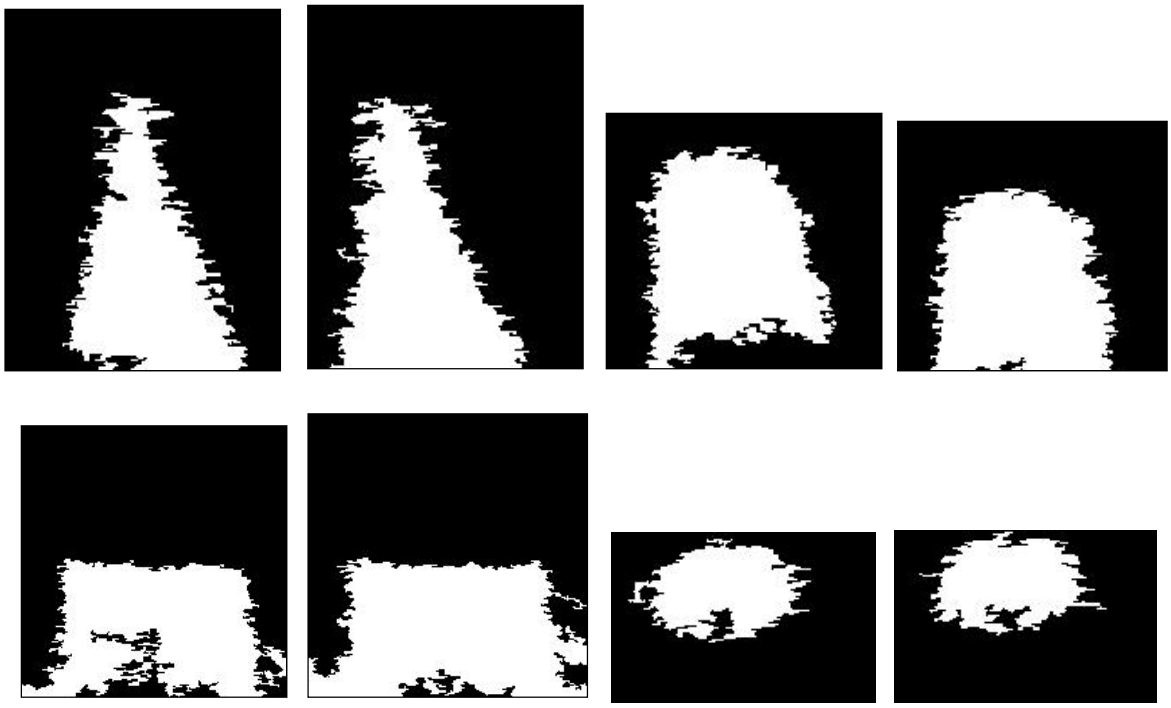
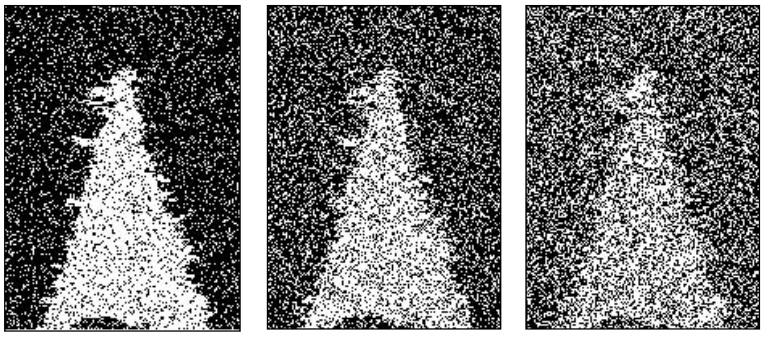
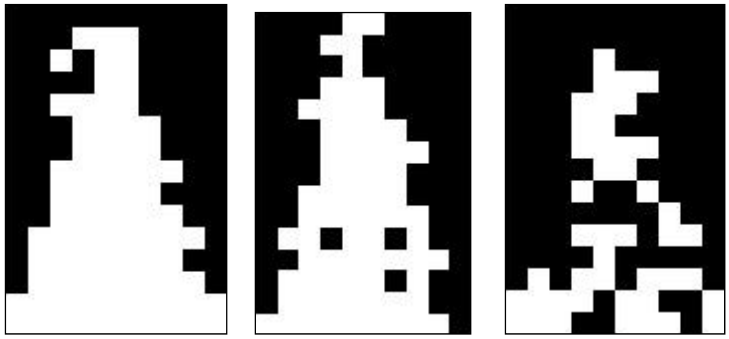


Fig. UOD-6. Test images used to confirm the BAM



(a) 30% noise (b) 50% (c) 70%



(d) 30% (e) 50% (f) 70%

Fig. UOD-7. Examples of noised image (a-c) and its sampled binary image (d-f)

Table UOD-1. Image classification results with various images and noise ratios

Image	Noise ratio		
	30%	50%	70%
T. Cylin. #1	OK	OK	Fail
T. Cylin. #2	OK	OK	Cylinder
T. Cylin. #3	OK	Fail	Fail
T. Cylin. #4	OK	OK	Cylinder
Cone #1	OK	OK	Fail
Cone #2	OK	Fail	Fail
Cone #3	OK	OK	Fail
Cone #3	OK	OK	Fail
Cylinder #1	Cube	OK	Cy. & Cu.
Cylinder #2	OK	Cube	OK
Cylinder #3	OK	OK	Cy. & Cu.
Cube #1	OK	OK	OK
Cube #2	OK	OK	OK
Cube #3	OK	OK	OK
Cube #4	OK	OK	OK

Mid-range localization: the DIDSON sonar – Research Approach 2

Underwater object recognition is the key to automate the underwater manipulation and tasks. 2D sonar and optical camera systems have been widely used for object recognition. However, several major disadvantages have hindered their practical implementation. 2D sonar's range is relatively long and reliable but higher resolution than what has been traditionally available is required for use in object recognition. Alternatively, optical camera systems have been shown to have the required resolution, but the limited visibility encountered in underwater environments restricts working range and reliability.

The high-resolution acoustic cameras such as DIDSON can be an alternative. It has the required range and reliability needed for optical recognition combined with the high resolution seen in traditional optical vision systems. However, the acoustic camera has unique characteristics that hinder its use in autonomous object recognition tasks. The acoustic camera system has several defining features that make image recognition more challenging when compared to a traditional optical camera system.

Due to the differences between the optical camera model and DIDSON model and conventional recognition technique are difficult to apply to the acoustic camera image. As a result, to develop an effective image recognition algorithm, an acoustic camera model and recognition method had to be developed that considered the acoustic reflection characteristics of the acoustic camera. In order to overcome the above difficulties and implement to SAUVIM AUV, we propose the acoustic camera, DIDSON, model and the recognition method.

The image obtained by the acoustic camera, DIDSON, is highly sensitive to the camera's position. The camera model allows us to predict the actual shape of the target object based on the image obtained for any given arbitrary position of the camera.

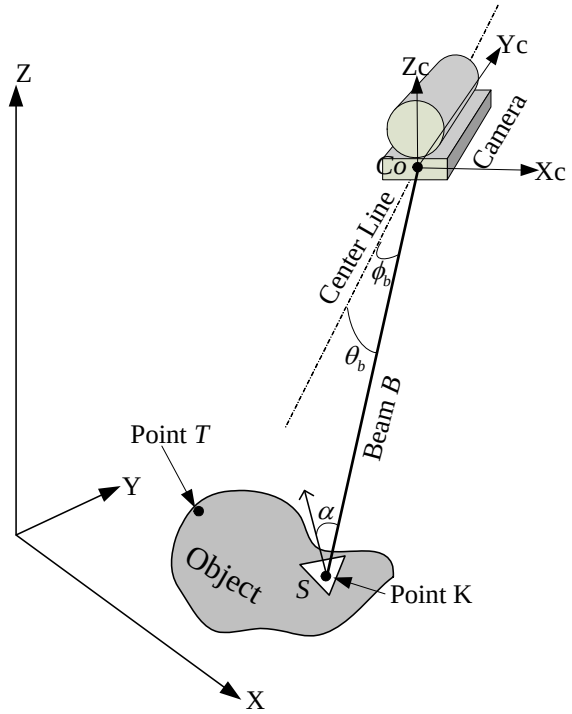


Figure AORD-7. Geometry for Acoustic Camera Model

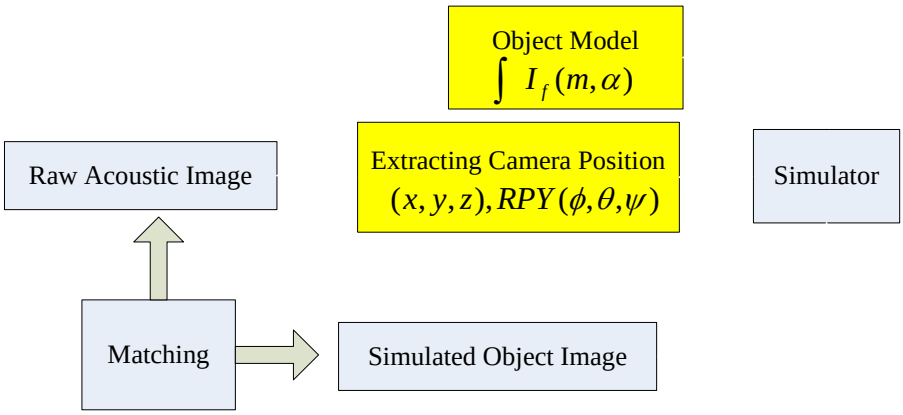


Figure AORD-8. Proposed recognition method

-DIDSON Modeling

The orientation of the camera beam can be determined by the two angles θ and ϕ, defined in the spherical coordinate system. As the camera tilts and pans (rotation on the XcZc and XcYc planes, respectively) the beams can be projected in varying directions and distance and intensity data collected. Discrete tilting and panning angles with regular increments are used in the actual implementation. The tilting of the acoustic camera generates a vertical beam slices consisting of *N* number of tilted beams. Each beam returns a distance *D* and intensity *I*. The distances and intensities are then mapped to a single line in the camera image. By panning the vertical beam a series of lines can be generated that is used to construct a whole image. Assume an object is located in the camera's visible range. Let a point *T*,as shown in Figure AORD-7, on the object, set the origin of the image. Let the position and rotation of *T* be known. A beam *B* hits the object surface at the point *K*. The beam *B*'s pan and tilt angle from the center line are ϕ_b and θ_b, respectively, and the length(range) of the *B* from point *C0* to point *K* is known. This describes the line *B*. Assume that the object's surface can be described the collection of triangular elements and the line *B* hits at the surface of element *S*.

The intersection of the *B* and *S* determines the point *K*. The distance *D* can be estimated using the distance between points *K* and *Co*. The intensity depends on object's surface composition and shape. The surface composition is, in turn, related with acoustic reflection characteristics *ma*, and the reflection angle α between the normal vector of the *S* and *B*. *I* can be estimated using the function $I_f(ma,\alpha)$ at *K*.

Using this method, each beam's distance *D* and intensity *I* can be estimated and used to construct the whole image frame as shown in Figure AORD-9. This enables to predict an object image in the camera at a certain point *T*.

When an object image is taken, the camera position and the rotation can be estimated. Since all beam's ϕ_b, θ_b and *D* is known, Tz, Troll and Tpitch can be estimated using triangulation. Due the large amounts of noise in the acoustic image, ellipse approximation has been found

to be the most efficient method of estimating the object's heading angle in the image. Using this camera model, the corresponding target image model can be used for the recognition.

Recognition Cues

Generally, the optical camera uses the bright area of an object as a cue to recognize the object. Most often, the shadow area of the object is not used.

In the acoustic camera's case, shadow areas generated turn out to be the more general and reliable cues that can be used to recognize the object.

In DIDSON images, bright areas that are easy to recognize are only generated under the limited conditions, such as when the there is a flat and coarse surface. However, all objects which have heights generate a shadow and generally most seabeds give good enough contrast to detect shadows.

Table 1 groups the available cues of objects based on the experiment results.

Recognition Method

As a recognition method we propose to use the objects shadow for recognition. By comparing an objects shadow with a predicted shape recognition can be made. Acoustic shadows are less dependent on acoustic refection, and as a result more stable and reliable. The shadow is recognized using the correlation of the actual and simulated shadows' X and Y Axis projection. Let correlation of X-axis projection between the model and the actual image be CorrXp and Y-axis be CorrYp and the length of the object at X and Y axis are Lx,Yx.

The object correlation value *Ct* can be expressed using the following equation:

$$Ct = K_1 \times CorrXp + K_2 \times CorrYp \quad (1)$$

$$K_1 = \frac{Lx}{Lx + Ly}, K_2 = \frac{Ly}{Lx + Ly} \quad (2)$$

The longer length of the shadow the longer projection values exist and the more information is provided. This method is robust to the edge and/or inner area noise and efficient enough to realize the real-time processing with small computing power.

Experiment

In order to estimate the accuracy of the camera model and the proposed recognition, experiments were carried out. The goal was to recognize 5 simulated models (brick, cube, cylinder, tapped cylinder, and cone) using the actual images as shown in Figure AORD-11. Tables 2 illustrate the recognition results. The object's image was taken by the DIDSON with regular interval and the different actual images of the same object were used at each table to test the changing feature effect.

All objects were successfully recognized. The correlation result, *Ct*, was highest for when the simulated model was matched with the actual image of the object in question. The cone and the tapped cylinder showed quite similar values. This was due to the similarities seen in their shadows. These results confirm the accuracy of the proposed camera model and recognition method.

When the object is not symmetric, the object's angle needs to be estimated and used with the correct model to generate a recognition method. The object's angle estimation experiment was carried out to evaluate its accuracy. The mentioned ellipse approximation method was used.

As shown in Figure AORD-12, an object, a brick was installed on the turn table and rotated from 0 to 360 degree. The brick's bright area was used for the angle estimation. Every 30 degree, the brick's image was taken and its angle was estimated. The simulated brick's angle was estimated in 1 degree intervals. Figure AORD-13 shows the result. The actual and simulated results showed the good agreement.

The estimated angles in Y axis were changed from -30 to 30 degree while the actual rotation in X axis changed from 0 to 360 degree. This phenomenon is due to the slant of brick with respect to the camera and can be compensated for using the geometry and knowing the slant angle. Around 270 degrees, a large error can be observed. At 90 and 270 degree, the brick's shape becomes very similar to a square when viewed at a slant. In this case, the ellipse approximation is difficult since the length of major and minor is similar.

This phenomenon can avoided by finding the ratio between the brick's X-axis projection length with respect to the Y-axis projection length to find these angles. As shown in Figure AORD-14, the X/Y ratio approaches 1.0 near of 90 and 270 degrees for the actual and simulated cases.

We proposed an acoustic camera model and recognition method for autonomous underwater manipulation. The recognition method considered acoustic characteristics of the DIDSON camera. As a result, high accuracy and reliability of this recognition method achieved and proved experimentally. Optical camera models enables prediction object shapes and shadows at certain points. This plays an important role object recognition. This recognition method allowed for reliable recognition with minimal computing power. Using the camera model and study of the intensity function, which related with object shape and material, a complicated object can be simulated with high accuracy. Side scan sonar's displaying mechanism is very similar to those generated by the DIDSON acoustic camera. The proposed camera model and shadow recognition algorithm can be applied to recognize side scan sonar images of object or seabed elevation maps.

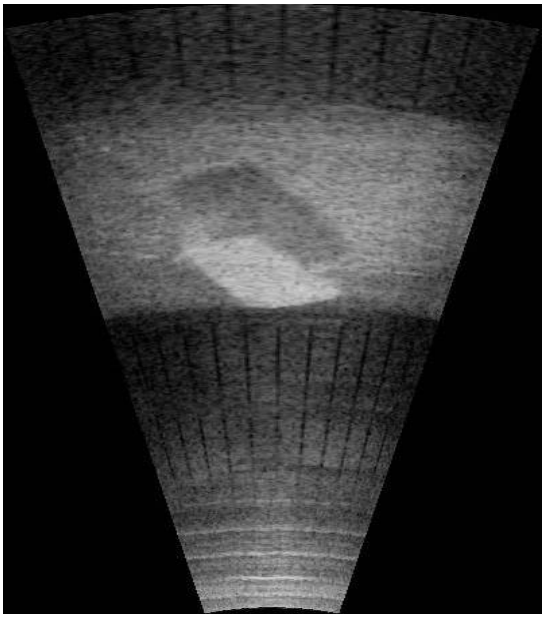


Figure AORD-9. Brick's raw image

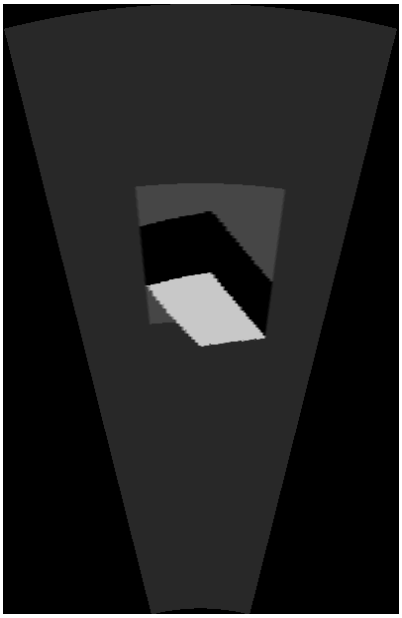


Figure AORD-10. Predicted image

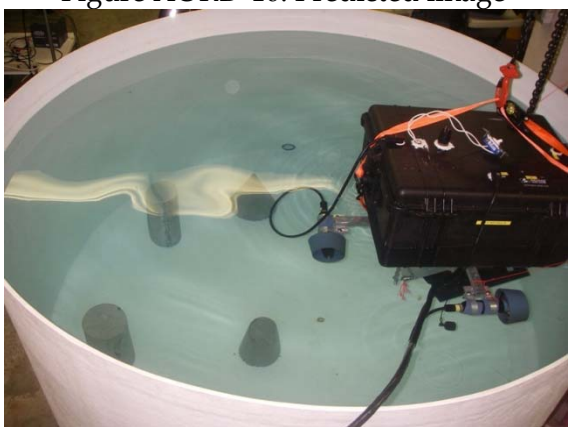


Figure AORD-11. Objects identification test

Model / Actual	Brick	Cube	Cylinder	Tapped Cylinder	Cone
Brick	**0.937**	0.871	0.814	0.811	0.786
Block	0.847	**0.979**	0.878	0.885	0.830
Cylinder	0.781	0.874	**0.982**	0.889	0.868
Tapped Cylinder	0.743	0.851	0.899	**0.986**	0.953
Cone	0.784	0.861	0.912	0.961	**0.989**

Table 2 Recognition Result2, Correlation Values

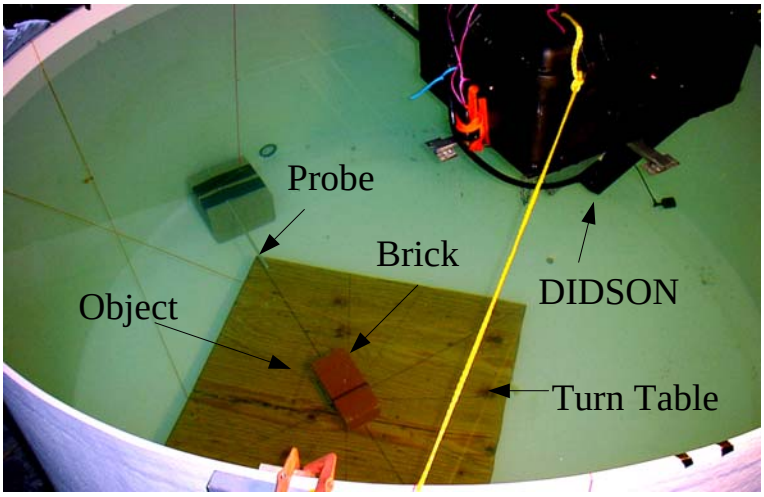


Figure AORD-12. Object Rotation Test Experiment Setup

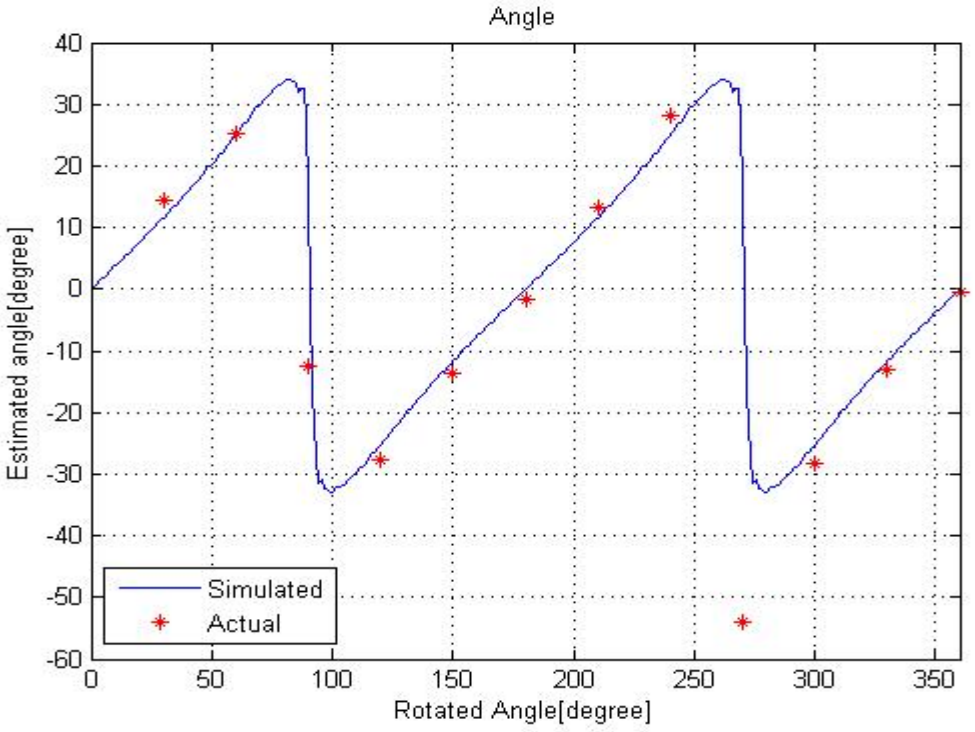


Figure AORD-13. Rotation Angle Estimation Result

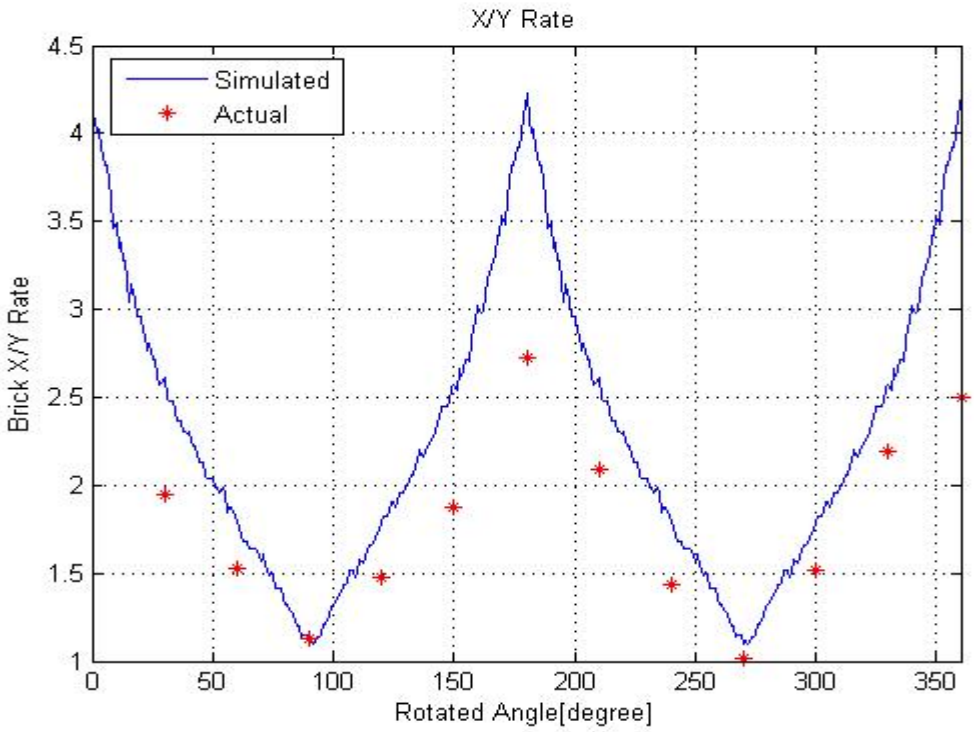


Figure AORD-14. Brick's X /Y –Axis Projection Length Ratio

Short-range target localization

When the target is within the manipulator workspace, short range and high accuracy sensor are used in order to perform the actual intervention task. This goal is achieved with the combined use of underwater video cameras and an ultrasonic motion tracker, used to retrieve the real-time 6 DOF position of the target during the manipulation tasks.
In this phase, a key feature of the autonomous manipulation system is the capability to locate the position of the target with respect to the base frame of the manipulator (which often may coincide with the vehicle frame) with a number of degrees of freedom sufficient to perform the required task.
In our task, the target is the dipole of Figure AORD-15. The two spheres composing the dipole are of known diameter.

Sphere localization using video processing

In Figure AORD-16, each sphere is represented schematically with the associated frame $< t >$. The transformation matrix T_t^0 of the target frame $< t >$ with respect to the base frame $< 0 >$ is given by:

$$T_t^0 = T_c^0 T_t^c \qquad (1)$$

where T_c^0 is the transformation matrix of the camera frame $< c >$ with respect to the base frame $< 0 >$, while T_t^c is the transformation matrix of the target frame $< t >$ with respect to the camera.
The placement of the camera T_c^0 is easily computed using the joint position information of the arm and the relative position of the camera with respect to the joint on which it is mechanically coupled. This relative position may be precisely computed using a set of predefined movement of the joint hosting the camera, along the main axes.

Figure AORD-15. The target in our recovery experiment

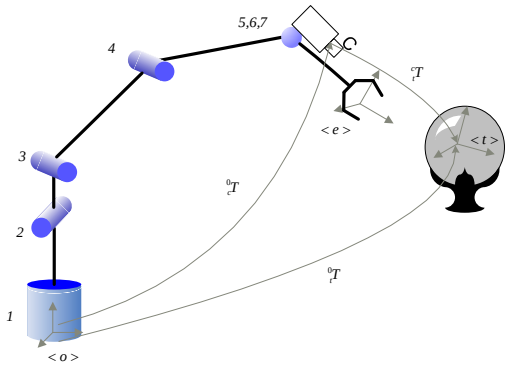


Figure AORD-16. Schematic representation of the arm workcell with camera and target.

The placement of each sphere location T_t^c with respect to the camera is obtained, instead, using video processing of the acquired image. As a matter of fact, the problem may be seen as a 2d localization of a circle within the acquired image. Upon camera calibration, the cartesian 3D position (x_s, y_s, z_s) of the center of the sphere in the space may be easily computed from the center and the diameter of the 2D circle, using the following relationships:

$$\begin{aligned} z_s &= K_z \frac{r_s}{r_p} \\ x_s &= K_x z_s \left(x_p - x_{p0} \right) \\ y_s &= K_y z_s \left(y_p - y_{p0} \right) \end{aligned} \qquad (2)$$

where r_s is the actual radius of the sphere, r_p is the measured radius in pixels, x_p and y_p are the position of the center of the circle within the acquired image (in pixels), x_{p0} and y_{p0} are a translation factor depending from the size of the image (in pixels).
The constants K_z, K_x and K_y are the calibration parameters and depend mainly by the focal length of the camera and the physical dimension of the pixels. Their values are computed directly in the water, with the aid of the robotic arm that makes possible the acquisition of a fixed optical pattern from different angles and distances.
The localization of the circle within the image is done using the following sequence of steps:

- Image filtering.
- Edge extraction using Canny filter applied to the color image and using the color contrast gradient.
- Circle extraction using the line segments found in digital images (Kim and Kitajima, 2005).

This combination of algorithms gave us the best performance and robustness with respect to false or missed detections in an underwater environment. Figure AORD-17 shows the result of the above sequence of steps applied to a single frame. In our actual implementation, the

system is capable of processing about 10 frames per second, which is sufficiently high in order to lock and follow the target, in case of a relative movement of the target with respect to the vehicle.

Figure AORD-17. Result of video processing: the circle extraction.

Estimation of the target position

The presented algorithm for preliminary sphere detection must be of course supported by a post data analysis in order to validate the target position. As a matter of fact, the results of the circle extraction may vary considerably according to several parameters, such as the visibility of even the presence of unwanted false detections. This is true, in the general case, for any kind of sensor data, and the solution here introduced for the estimation of the target position and velocity may be regarded as a general procedure for target tracking.

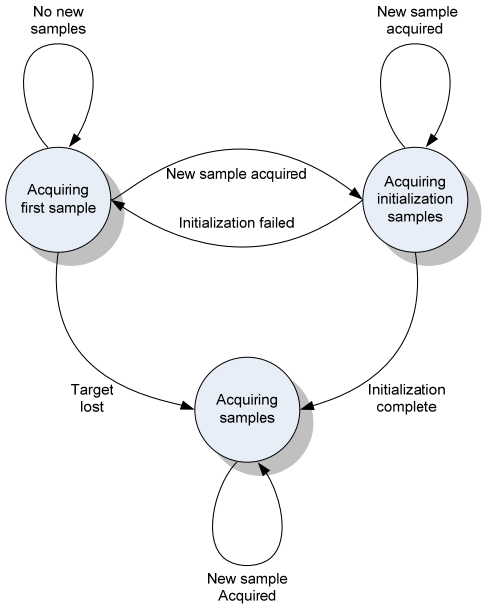


Figure AORD-18. The finite state machine for a single target validation.

The post-analysis has been implemented using the finite state machine schematized in Figure AORD-18 for each feature to be detected. The algorithm starts by analyzing a single frame and collecting all the detected features (spheres in our case). This phase, which involves the previous algorithm for 3D sphere detection, builds an array of several 3D positions triplets, each of one describes the position of a different feature (sphere) detected in the space. The number of features detected may vary from zero up to a defined maximum (4 in our implementation). Each sample is sent to the input of a separate finite state machine, according to the following matching rule.
The set of positions of the features detected is associated to the correspondent state machine according to a rule depending by the number of samples acquired within the FSM. If only one sample has been acquired, each feature of the successive frame is associated to the FSM if the position difference with respect the only sample acquired is less then a predefined threshold. In any other case, the data collected are matched with the values of an interpolation functions, extrapolated in the future. The interpolation function is computed by finding the least-square solution of a linear combination of a given base of functions. In our choice, we found optimum results with a polynomial:

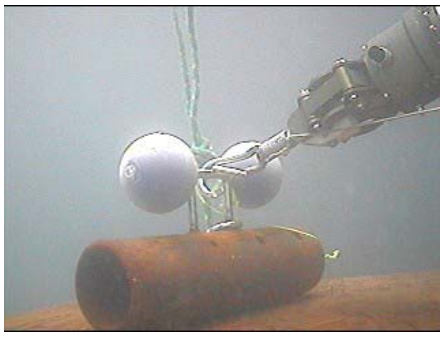
$$y(x) = a_1 + a_2 x + a_3 x^2 + \ldots + a_m x^{M-1} \qquad (3)$$

The least-square solution is the set of coefficients which minimizes the merit function:

$$\chi = \sum_{i=1}^{M} \left[\frac{y_i - y(x_i)}{\sigma_i} \right]^2 \qquad (4)$$

The solution is computed using the singular value decomposition technique.
In the initialization state of the FSM of Figure AORD-18, the *Initialization failed* event is generated upon a single sample lost. Instead, in the *Acquiring samples* state, the *Target lost* event is generated if there are more that a predefined number of no detections in sequence. The initialization complete event not is generated until a predefined number of features (in our case 5) are validated. From now on, the interpolation function is built excluding the oldest sample and adding the newest one, which is matched against the previous-built interpolation function (extrapolated in the future). At this stage, the measurement error σ_i of Eq. 4 is computed in order to give more weight to the newest samples.
This solution allows a better estimation and validation of the target position, providing a continuous tracking even in the case that the target is temporarily occluded by the environment. **Error! Reference source not found.** shows the scatter plots of the position error (in pixels) of the sphere along the camera abscissa, respectively for $M = 2$ and $M = 3$ in Eq. 3. The data samples have been collected over the same movement of the camera, and the improvement using a more informative estimation function is noticeable.
In our case, the target is composed of 2 spheres joined by a connecting rod. Hence, the final validation occurs if there are at least 2 spheres validated and also if they are at a set relative distance from each other with respect to the original dipole length and their distance from the camera. All the other detections are discarded.
The results are extremely accurate, with zero false detections and an accuracy close to 2% of the distance of the dipole from the camera (the resolution if the image, in our case, was 320x240 pixels).

(a) Searching for the target



(b) The manipulator in the hooking phase

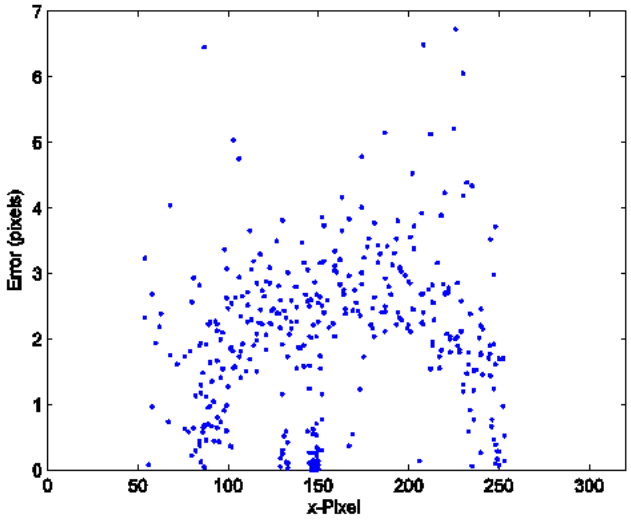
Figure AORD-19. Underwater scenes of the target recovery task.

Application example

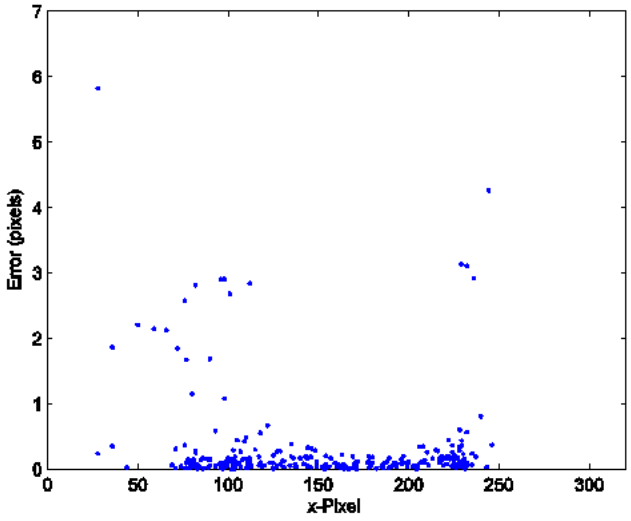
The presented solution for autonomous underwater manipulation has been validated within an intervention mission using SAUVIM and is presented here. The mission consists in a recovery operation of the submerged target of Figure AORD-15, using the arm in order to securely clamp to the target an underwater inflatable lift bag.
The entire sequence of operation involved in this demo has been coded within the High Level Controller layer and consists in the following subtasks:

- Extract the arm and perform a visual scan (in 3D) of the surrounding space, using the attached camera (**Error! Reference source not found.**a).
- During the scan, try to locate the target.
- Once the target has been detected, the arm enters in a tracking state, in order to place the gripper to a constant relative position with respect to it. If the target moves, the arm follows it while maintaining the same relative position.
- Once the tracking system detects no movements of the target for a sufficient amount of time, the arm proceeds with the short sequence of movements finalized to hook the snap link (**Error! Reference source not found.**b). During the time frame of the hooking operation, any movement of the target with respect to the arm may still be corrected using the video feedback.

The only required human intervention is the decision on when starting the initial search. As a matter of fact, the problem of long-range searches involves different technologies and is beyond the scope of this paper. Once the vehicle has reached the target site, it is matter of the supervisor to decide if the target is the correct one and, successively, to start the autonomous sequence of operation. From now on, the arm performs autonomously all the subsequent operations in order to reach the recovery goal. If any errors arises during any of the above autonomous steps (for example no target has been detected during the scan), the arm returns to its initial parking position and the mission is aborted. A mission log allows the operator to verify later the cause of the failure.



(a) First degree polynomial



(b) Second degree polynomial

Figure AORD-20. Scatter plots of the position error of the sphere along the camera abscissa (in pixels).

Ultrasonic Motion Tracker

Problem statement: Short Range Underwater Target Localization

Some sensors, as for example video processing and laser or ultrasonic 3D scanners, provide an absolute measurement, even if usually with a low sample rate and high cost. Video processing, however, may present some drawbacks in the ocean depths. The need of a constant light source during the manipulation task may considerably degrade the autonomy of the vehicle. Moreover, the poor visibility in some environments may introduce some difficulties in the target detection/recognition process.
On the other hand, an ultrasonic motion tracker can provide reliable and high sample rate information, but the measurement is relative to the position of the probe with respect to the target. This means that the system must know exactly the point of application of the ultrasonic receiver (the sensor probe). However the sensor application/localization has to be done only once, and can be achieved substantially in one of the following way:

- *Operator assisted*. In case of a sufficiently reliable link, the application of the sensor to the target can be executed by the operator using teleoperation and/or teleprogramming mode (Sayers, Paul, Catipovic, Whitcomb and Yoerger, 1998). This is sometime referred as a semi-autonomous modality of execution of the task.
- *Autonomous mode*. The application of the probe to the target is executed in autonomous mode with the aid of the above mentioned absolute measurement 3D sensor (video processing, scanners…).

After this phase, the manipulation task can be executed using only the information of the motion tracker.
The motion tracker-aided manipulation is conceptually similar to the use of passive arm measurement devices. The main advantage is the absence of a mechanical link between the target and the AUV, which becomes a simple wire or even absent in case of wireless sensors.

Commercial Solutions

The commercially available motion trackers are based on the following three different technologies.
We can refer to the first one as "magnetic tracking" (Raab, Blood, Steiner, and Jones, 1979, Paperno, Sasada and Leonovich, 2001): this kind of trackers are used to capture translation coordinates [x,y,z] and yaw, pitch, roll [y,p,r] rotation coordinates. The transmitter consists of three coils on orthogonal [x,y,z] axes, with an excitation current (either AC or DC) passing through each coil. The sensor consists of a similar set of three coils. Unfortunately, this device has several drawbacks (field distortion and interference, distance diminishes accuracy, latency and jitter) and such technology cannot be used for precision underwater manipulation tasks.
Another solution is ultrasonic tracking (Mahajan and Walworth, 2001): an ultrasonic tracker utilizes high frequency sound waves to track objects by either the triangulation of several transmitters (time-of-flight method) or by measuring the signal's phase difference between transmitter and receiver (phase-coherence method). The "time-of-flight" method of ultrasonic tracking uses the speed of sound through air to calculate the distance between the transmitter of an ultrasonic pulse and the receiver of that pulse. The use of at least 1 transmitter at the stationary positions and 3 receivers on the tracked object allows the

system to determine the relative cartesian position (3 DOF) of the object via triangulation. The use of at least 3 transmitters at stationary positions with 3 receivers on the target can be used to determine the position and orientation (6 DOF) of the object.
The "phase-coherence" tracking method estimates the target location by sensing the signal phase difference between the signal sent by the transmitter and the one detected by the receiver. The phase-coherent tracking is an incremental form of position determination, and small errors in position determination will result in larger errors over time (drift errors).
Some weak points of ultrasonic trackers are associated to the line-of-sight requirement of the transmitter and receivers. This requirement plagues the tracker with shadowing problem and limits their effective tracking range. They are also very susceptible to interference caused by reflections of the ultrasonic signals from hard surfaces and interference from ambient noise sources.
Another tracking technology is the inertial tracking (Lang, Kusej, Pinz and Brasseur, 2002, Mazl and Preucil, 2003). The general principles in inertial tracking are to measure the accelerations (accelerometers) or the orientation (gyroscopes). Several technologies are available today for acceleration measurements, as for example the Fiber Optics gyroscopes and the MEMS (Microelectromechanical systems) accelerometers. In any case, inertial tracking is based on integration, which causes the actual positions and orientations to be sensitive to drift, and have to be re-calibrated periodically.
Finally, hybrid tracker technologies combines the previous described ones. This technology has been investigated in Mazl, Preucil, 2003, Suya, Neumann and Azuma, 1999, McCarthy, Duff, Muller and Randell, 2006.

Proposed Approach

The commercially available motion trackers are mostly developed for virtual reality purposes (for instance in capturing the body movements) or medical use (i.e. for tracking the position of probes). However, the underwater environment lacks of devices for high definition measurement of generalized position (translation and rotation) to be employed in robotic tasks.
Hydroacoustic Position Reference systems (HPRs) are set to provide positioning information mostly for navigation purpose, with an accuracy targeted to the requirements of the navigation task. HPR systems include Ultra- or Super- Short Base Line (USBL or SSBL), Long Base Line (LBL) and Short Base Line (SBL). While the information provided by the above system is generally excellent for navigation purpose, it is usually insufficient to measure the position of a target for a robotic intervention task. In fact, the most distinguishing features required in an underwater robotic intervention are:

- Accuracy. Generally a robotic task may require a high degree of conformity of a measured quantity to its actual value, often in the order of millimeter.
- Information. A robotic task requires the knowledge of the full 6 DOF generalized position (rotation and translation) of the target with respect to the main frame (HPR systems often provide only Cartesian position).
- Size. The measuring probe must have small size in order to avoid interaction issues with the target.

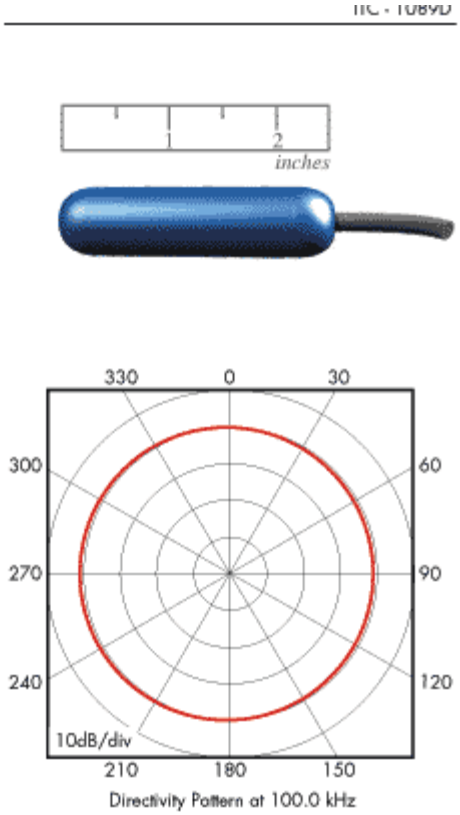


Figure AORD-21. Underwater ultrasonic transducer ITC-1089D (from International Transducers Corporation).

Our goal is to address the above issues, using the available technology in order to cope with the requirements of a generic autonomous underwater manipulation task. This underwater tracking technology can be also used in different situation as for example in precision vehicle docking/undocking procedures (Evans, Redmond, Plakas, Hamilton and Lane, 2003). While the metal structure of an AUV, as well as of the target, would suggest avoiding magnetic devices, the nature of the environment is suitable for the use of the ultrasonic technology.
One of the key devices in underwater ultrasonic tracking is the transducer. Its choice must be done accordingly with a reasonable balance between power, depth, bandwidth and cost. The International Transducers Corporation (www.itc-transducers.com) offers a product suitable for our application (Figure AORD-21): the characteristics of large bandwidth, compact size and depth range are the best compromise for our tracker requirements.

Localization in one dimension

The most critical and fundamental step in our development is the measurement of the distance between the transmitting and receiving transducers.
This is accomplished, in our approach, in two phases.
The first consists in measuring the delay δ_t between the transmitted and received waveforms (Figure AORD-22).

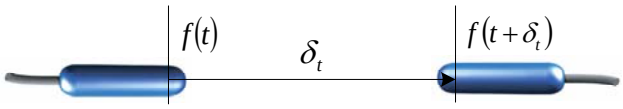


Figure AORD-22. Transmitting and receiving transducers pair.

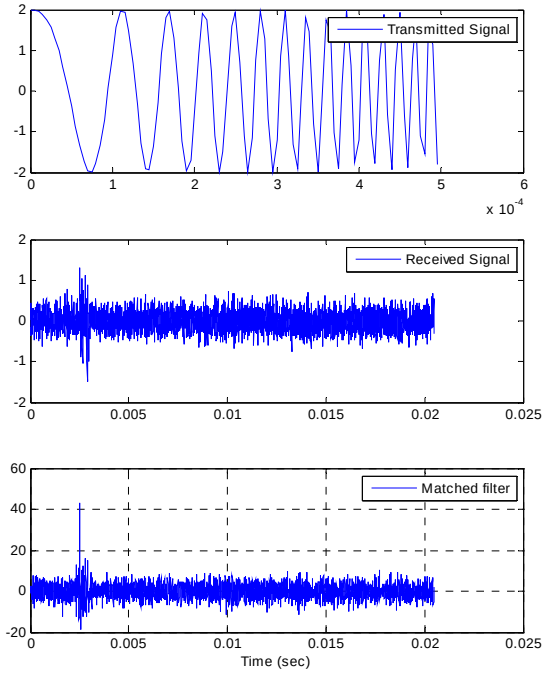


Figure AORD-23. Signal detection with matching filter (simulation).

The temporal localization of the received waveform $f(t)$ is performed using a matched filter, for maximizing the signal-to-noise ratio. The transmitted waveform has been chosen in order to increase the spatial localization after the matched filter. We use a slightly modified Linear Frequency Modulated Chirp (LFM Chirp):

$$s(t) = K \cdot cos\left(2\pi\left(f_0 + \frac{f_{max} - f_0}{\Delta T}t\right)t\right) \quad (1)$$

The frequency of the chirp signal sweeps from f_0 to f_{max} over a period ΔT. It is interesting to note that the phase of $s(t)$ varies quadratically versus t while the frequency changes linearly versus time. The derivative of phase determines the instantaneous frequency of the signal. With this assumption, the matched filter for the above signal is (in the frequency domain):

$$h(f) = S^*(f) \quad (2)$$

where $S^*(f)$ is the complex conjugate of the Fourier transform of the transmitted signal $s(t)$.

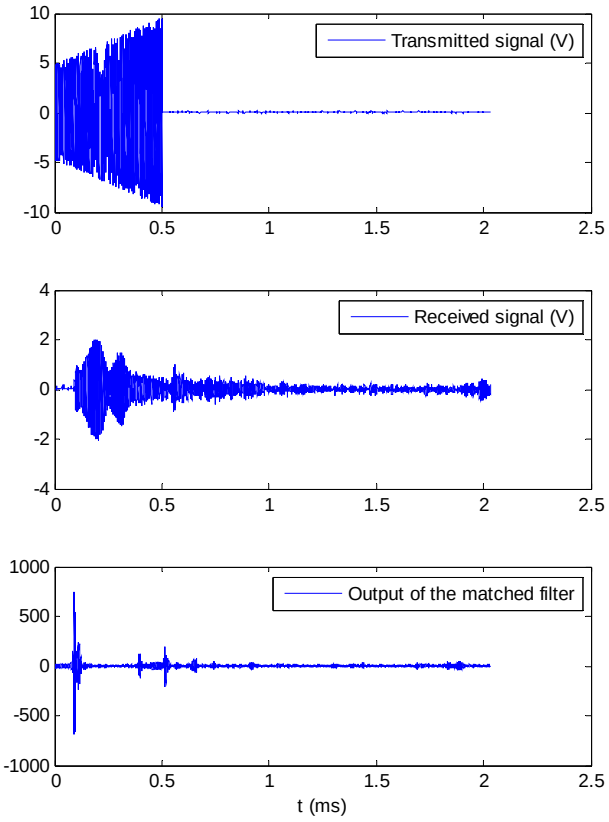


Figure AORD-24. Signal detection with matching filter (experimental results).

Figure AORD-23 shows a simulative example of the above filter, applied to a signal with $f_0 = 3kHz$, $f_{max} = 30kHz$ and $\Delta T = 0.5ms$.
To show the performance of the filter we added some Gaussian noise (with a variance of 0.05) to the received signal. Even if extremely noisy, the matching filter is still able to successfully detect the location in the time of the original waveform.
Moreover, as shown in Figure AORD-23, the use of a LFM chirp leads to greater localization of the wave in the time domain, which is desirable in our application.

Hardware implementation

The tracker unit has been implemented as in Figure AORD-25, where the signals to and from the transducers (ITC 1089-D) are digitalized using a simultaneous sampling AD/DA board from General Standards Corporation, and finally processed using a Intel-based PC104 computer.
We tested the above system in one-dimension localization, using $f_0 = 250\,kHz$, $f_{max} = 350\,kHz$ and $\Delta T = 0.5\,ms$, with a sample frequency of $2\,MHz$ in reception and $1\,MHz$ in transmission. The transducers where submerged in clear water at a distance of about $135\,mm$.

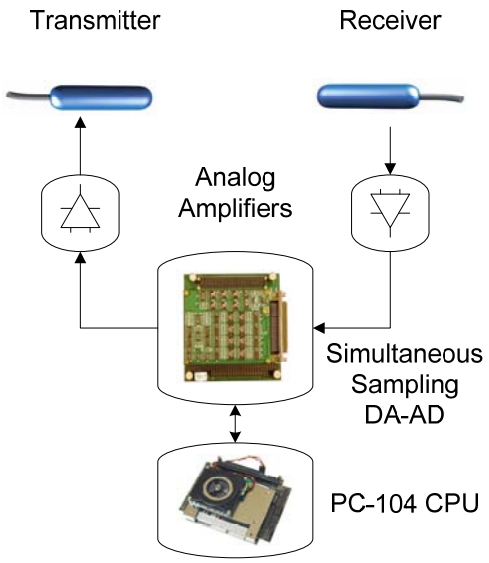


Figure AORD-25. Hardware implementation of the tracker (only one channel is shown).

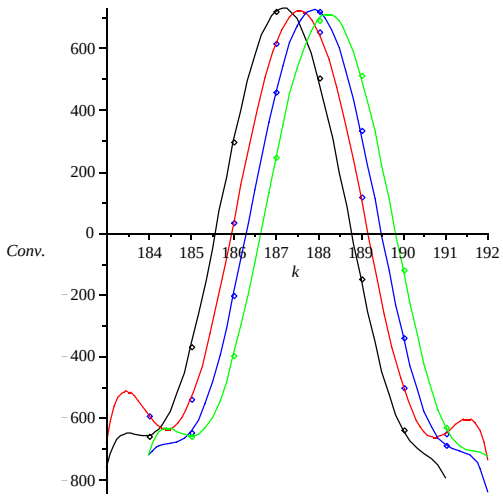


Figure AORD-26. Interpolation of the peak samples.

Results are shown in Figure AORD-24. The variation of the amplitude of the transmitted chirp is computed in order to compensate the variation of the transmitted power with the frequency. As in the simulative case, the matching filter is performing well within our level of noise, generated by the environment and by the reflections from the walls of the tank.

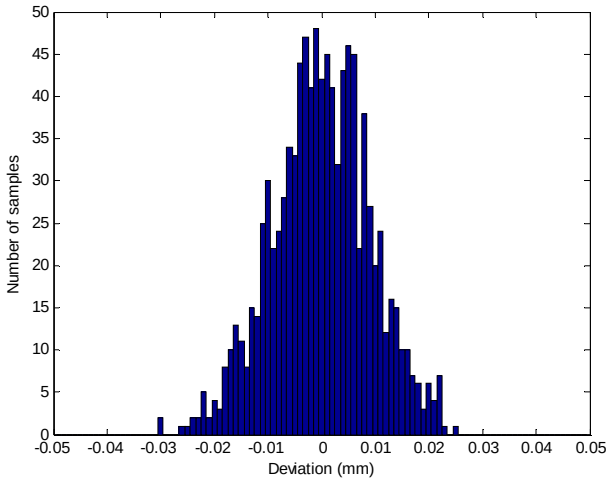


Figure AORD-27. Distribution of 1000 measurements around their mean values.

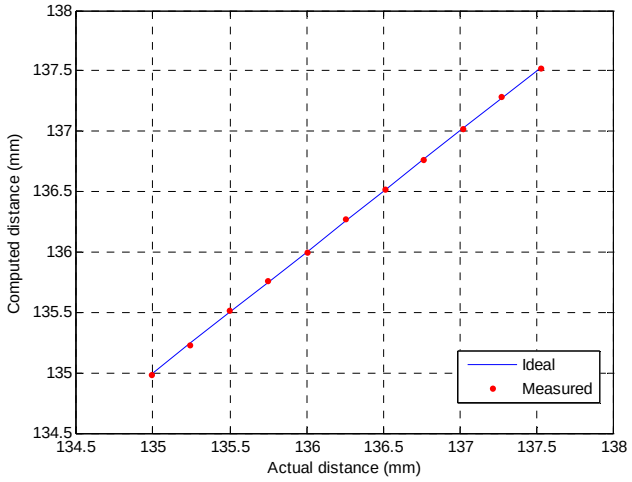


Figure AORD-28. Linearity of measures with a step increment of 1/1000''.

Peak detection

With this basic configuration, the spatial resolution (in time) of the peak is limited by the sample time of the received signal. A frequency of 2 MHz allows a time resolution of 0.5 μs, which corresponds to a distance of about 0.75 mm (in water).

In order to maximize the resolution, the second phase of the distance measurement consists of an improved algorithm for determining the location in the time domain of the peak vertex.

The algorithm, initially tries to validate a certain numbers (3 in our case) of peaks. This is accomplished only if the first peak is at least 50% higher of the last one. Then, in order to avoid false detections from reflections, only the first peak (in the time scale) is considered.

The value of the center of the peak is found by fitting the peak waveform with a sixth-degree polynomial.
The interpolation function is computed by finding the least-square solution of a linear combination of the given base of functions:

$$y(x) = \sum_{i=0}^{6} a_i x^i \tag{3}$$

which is a 6-th degree polynomial.
The least-square solution is the set of coefficients which minimizes the merit function:

$$\chi = \sum_{i=0}^{6} \left[\frac{y_i - y(x_i)}{\sigma_i} \right]^2 \tag{4}$$

where the weights σ_i are set to be all 1.
Since the degree of the (3) is even, it is possible to find the vertex of the polynomial using the last two coefficients:

$$t_p = -\frac{1}{6}\frac{a_5}{a_6} \tag{5}$$

Figure AORD-26 shows the application of the above principle to four different measurements taken at a distance step increment of $1/100''$ ($0.254\ mm$).
The results were excellent, allowing us to improve the accuracy of the position up to few hundredths of millimeter. This is confirmed by the statistical plot of Figure AORD-27, where we show the distribution of a 1000 samples acquisition of a steady value (translated to zero for clarity).
Finally, Figure AORD-28 gives one idea on the precision (and linearity) of the measure. Here, the samples where acquired placing the sensor on a mechanical device capable of measuring the distance step *increment* with an accuracy of $1/1000$ of inch ($0.0254\ mm$). Each value has been acquired averaging 1000 samples. Even in this case the precision and linearity were confirmed by our experimental results.
In all the above experiments, the range information has been computed from the time delay (5) and the speed of sound in water:

$$r = t_p \cdot c \tag{6}$$

where c is about 1531 m/s in sea water (Ulrich, 1967). For increased precision, the latter is computed using two of the transmitting transducers, one of which used as received. Since the relative distance is known, the sound speed at the working conditions (e.g. temperature and pressure) can be easily computed with a relative error given by:

$$\frac{\epsilon_c}{c} = \frac{\epsilon_d}{\delta_{ij} + \epsilon_d} \tag{7}$$

where ϵ_d is the absolute error on the distance measurement. In our experiment, with a fixed distance of 135 mm and an absolute error of $3 \cdot 10^{-5}\, m$, the speed c was found to be $1473 \pm 17\ m/s$.

Localization in 6 DOF

The extension to the multi-dimensional case is easily achieved using some geometrical considerations.
Figure AORD-29 shows the minimum configuration ($N = 3$) necessary to track a target in 6 degrees of freedom. The goal is to compute the position and orientation of the frame $< T_t >$ with respect to the main frame $< T_0 >$, given the N^2 ranges $r_{ij},\ i,j = 1..N$ measured with the above technique. For simplicity, Figure AORD-29 shows only the 3 ranges from the transmitter T1 to the receivers R1, R2 and R3.
The first (and most important) step in the localization consists in determining the coordinates of the receivers using the range measurements r_{ij}. Geometrically this corresponds to compute, for each transmitter, the intersection of the N spheres centered on the position of the N receivers, which leads to the following non-linear system of N^2 equations:

$$\left\{\left(x_{ti} - x_{rj}\right)^2 + \left(y_{ti} - y_{rj}\right)^2 + \left(z_{ti} - z_{rj}\right)^2 = {r_{ij}}^2, \qquad i,j = 1..N \right. \tag{8}$$

where (x_{ri}, y_{ri}, z_{ri}) are the coordinates of the i-th receiver and (x_{ti}, y_{ti}, z_{ti}) are the coordinates of the i-th transmitter.
Since it is not possible to avoid intrinsic measurement errors, an exact solution of the equations system (8) does not exist. Instead, the triangulation problem can be solved using the least-square solution which minimizes the function:

$$\min_{R_i \epsilon \Re^3} \sum_{i,j=1}^{N} \frac{1}{\sigma_{ij}} \left(r_{ij} - \left\|T_i - R_j\right\|\right)^2 \tag{9}$$

where T_i and R_i are the position vectors of the transmitters and receivers respectively, and σ_{ij} the variance of the gaussian noise associated with the measure r_{ij}.

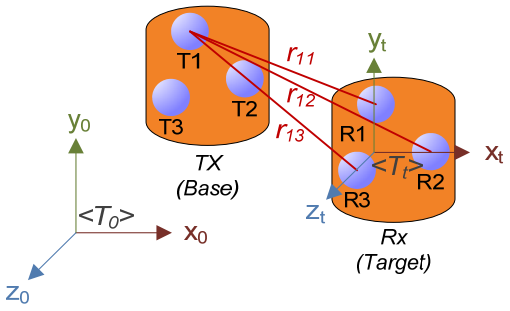


Figure AORD-29. Multidimensional case with 3 transmitters and 3 receivers.

This problem can be addressed by any of a number of computational methods (see Bancroft 1985; Misra and Enge 2001).
In our case, the solution is based on the Levenberg–Marquardt method (Madsen, Nielsen and Tingleff, 2004). To improve the accuracy and the convergence speed, we added some constrains based on the relative distance between transmitters:

$$\min_{R_i \epsilon \Re^3} \left[\sum_{i,j=1}^{N} \frac{1}{\sigma_{ij}} \left(r_{ij} - \left\| T_i - R_j \right\| \right)^2 + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(k_{ij} - \left\| R_i - R_j \right\| \right)^2 \right] \tag{10}$$

where k is the relative distance between two receivers.
In our experiment, since we used 4 receivers in a tetrahedral configuration, we have:

$$k_{ij} = K_t \tag{11}$$

where K_t is the length of the edge of the tetrahedron.
The simulative results of the above algorithm were converging successfully to the target values.
Finally, the translation and rotation of the target the frame $< T_t >$ can be easily computed from the above coordinates of the receivers, using simple geometrical transformations.
Any added transmitter-receiver pair has the effect of increase the precision of the result. In our approach, as introduced above, we plan to reach the final configuration with 4 transmitters and 4 receivers, the latter placed at the vertex of a tetrahedron structure in order to cover the full space.

Application example

This section shows a preliminary experimental result performed with a robotic manipulator, in order to validate the feasibility of the ultrasonic tracking during autonomous manipulation.
In this experiment, we used a commercial unit in air and the robotic manipulator of SAUVIM. The experiment consists in pouring the content of a test-tube in a container. The system (test-tube seat and container, Figure AORD-30) was prepared in a moving base, whose position was tracked by the ultrasonic tracking system. The small white triangle attached to the base is the receiver probe, while the transmitting unit (not shown in the pictures) was attached to the fixed structure (main frame) of the arm. The position and orientation of the tube and the container with respect to the ultrasonic probe were known by the robot before beginning the test.
During the above experiment a very important requirement was the accuracy of the absolute measure given by the tracker as well as the one of the end-effector computed from the joint angles. This was possible only after a precise calibration of the following quantities:

a. offsets of the joint angles
b. position of the (fixed) transmitter with respect to the arm
c. position of the receiver with respect to the test-tube

The above calibration was performed automatically by the arm using a set of particular movements finalized to identify all the missing parameters.

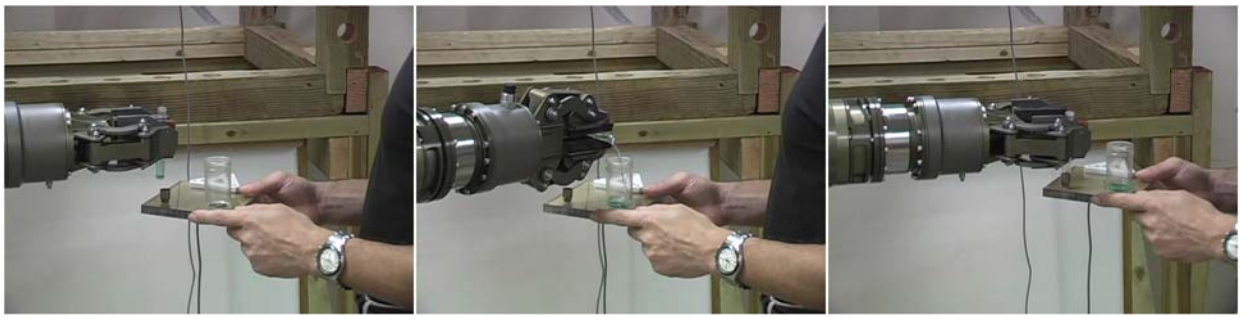


Figure AORD-30. Manipulation with moving target using the ultrasonic tracker.

Future Tasks (Phase III-C Tasks)

- New pattern recognition algorithm will be developed for DIDSON image. It will include noise reduction, isolated small blob removal, and morphological image processing for acoustic image. And, in order to increase reliability of image identification, neural network-based pattern recognition will be investigated and implemented.
- The DIDSON object detection system will be linked to the navigation controller
- Wet test will be performed for homing sensor system with image processing system. It will also confirm robustness of image processing algorithm in various light conditions as well as various turbidity conditions in the ocean.
- The motion tracked will be extended to 6DOF

References

[1] W. Fox, J. B. Hsieh, C. Polwarth, "Segmentation of images from an acoustic lens sonar," Proc. *MTS/IEEE OCEANS*, pp.2029-2034, Nov. 2004

[2] K. Haines, R. Hecht-Nielsen, "A BAM with increased information storage capacity," Proc. *IJCNN*, vol.1. pp.181-190, July 1988 (San Diego, CA, USA)

[3] B. Kosko, "Bidirectional associative memories," *IEEE Trans. System, Man, and Cybernetics*, Vol. 18, pp.49-60, Jan/Feb 1988

[4] B. Kosko, *Neural networks and fuzzy systems*, Prentice-Hall Inc., 1992

[5] Y. Lu, E. Sang, "Underwater target's size/shape dynamic analysis for fast target recognition using sonar images," Proc. *Underwater Technology*, pp.172-175, April 1998 (Tokyo, Japan)

[6] Y.-F. Wang, J. Cruz, J. Mulligan, "Two coding strategies for bidirectional associative memory," *IEEE Trans. Neural Networks*, Vol. 1, No.1, pp.81-92, Mar. 1990

[7] Y.-F. Wang, J. Cruz, J. Mulligan, "On multiple training for bidirectional associative memory," *IEEE Trans. Neural Networks*, Vol. 1, No. 3, pp. 275-276, 1990

[8] Y.-F. Wang, J. B. Cruz, J. H. Mulligan, "Guaranteed recall of all training pairs for bidirectional associative memory," *IEEE Tran. Neural Networks*, Vol. 2, No. 6, pp. 559-567, Nov. 1991

[9] S. C. Yu, T. W. Kim, G. Marani, S. K. Choi, "High resolution acoustic camera model based object recognition for AUVs", Proc. *UUST'07*, Aug. 2007 (Durham, NH, U.S.A.)

Aguilar, RN, Kerkvliet, HMM, and Meijer, GCM (2005). "High-Resolution Low-Cost Ultrasonic Tracking System for Human-Interface Systems", *Instrumentation and Measurement Technology Conference, IMTC 2005, Proceedings of the IEEE*, Volume 2, 16-19 May 2005, pp. 878-882.

Bancroft, S (1985). "An algebraic solution of the GPS equations", *IEEE Trans. Aerospace Electr. Syst*., AES-21(7).

Evans, J, Redmond, P, Plakas, C, Hamilton, K, Lane, D (2003). "Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation", *OCEANS 2003. Proceedings*, 22-26 Sept., Vol. 4, pp 2201- 2210.

Lang, P, Kusej, A, Pinz, A, and Brasseur, G (2002). "Inertial tracking for mobile augmented reality", *Instrumentation and Measurement Technology Conference, IMTC/2002, Proceedings of the 19th IEEE*, Volume 2, 21-23 May, pp 1583-1587, vol. 2.

Mahajan, A, and Walworth, M (2001). "3D position sensing using the differences in the time-of-flights from a wave source to various receivers", *Robotics and Automation, IEEE Transactions on*, Volume 17, Issue 1, Feb., pp 91-94

Madsen, K, Nielsen, HB, and Tingleff, O (2004). *Methods for Non-Linear Least Squares problems*, 2nd Edition, Informatics and Mathematical Modeling, Technical University of Denmark, April.

Mazl, R, and Preucil, L (2003). "Sensor data fusion for inertial navigation of trains in GPS-dark areas", *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*. 9-11 June, pp 345-350.

McCarthy, M, Duff, P, Muller, HL, and Randell, C (2006). "Accessible Ultrasonic Positioning", *University of Bristol Published by the IEEE CS and IEEE ComSoc*, pp 1536-1268.

Misra, P, Enge, P (2001). "*Global positioning system: signals, measurements, performance*". Ganga-Jumana Press, Lincoln, MA

Paperno, E, Sasada, I, and Leonovich, E (2001), "A new method for magnetic position and orientation tracking," *IEEE Trans. Magn.*, vol. 37, pp. 1938–1940, July.

Paul, RP, Sayers, CP, and Stein, MR (1993). "The theory of teleprogramming," Journal of the Robotics Society of Japan, vol. 11, no. 6, pp. 14–19, September.

Raab, FH, Blood, EB, Steiner, TO, and Jones, HR (1979). "Magnetic position and orientation tracking system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 5, pp. 709–717.

Sayers, CP, Paul, RP, Catipovic, J, Whitcomb, LL, and Yoerger, D (1998). "Teleprogramming for subsea teleoperation using acoustic communication", *IEEE Journal of Oceanic Engineering*, vol. 23, no. 1, pp. 60–71.

Suya You, Neumann, U, Azuma, R (1999). "Hybrid inertial and vision tracking for augmented reality registration", *Virtual Reality, 1999. Proceedings., IEEE*, 13-17 March, pp 260-267.

Ulrich, RJ (1967). "Principles of Underwater Sound for Engineers", *McGraw-Hill*, NY.

Yuh, J, and Choi, SK (1999). "Semi-Autonomous Underwater Vehicle for Intervention Mission: An AUV that does more than just swim", *Sea Technology*, (40) 10:37-42.

Yuh, J, Choi, SK, Ikehara, C, Kim, GH, McMurty, G, Ghasemi-Nejhad, M, Sarkar, N, and Sugihara, K (1998). "Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM)", *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on*, 15-17 April, pp 63-68.

Aguilar, RN, Kerkvliet, HMM, and Meijer, GCM (2005). "High-Resolution Low-Cost Ultrasonic Tracking System for Human-Interface Systems", *Instrumentation and Measurement Technology Conference, IMTC 2005, Proceedings of the IEEE*, Volume 2, 16-19 May 2005, pp. 878-882.

Bancroft, S (1985). "An algebraic solution of the GPS equations", *IEEE Trans. Aerospace Electr. Syst.*, AES-21(7).

Evans, J, Redmond, P, Plakas, C, Hamilton, K, Lane, D (2003). "Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation", *OCEANS 2003. Proceedings*, 22-26 Sept., Vol. 4, pp 2201- 2210.

Lang, P, Kusej, A, Pinz, A, and Brasseur, G (2002). "Inertial tracking for mobile augmented reality", *Instrumentation and Measurement Technology Conference, IMTC/2002, Proceedings of the 19th IEEE*, Volume 2, 21-23 May, pp 1583-1587, vol. 2.

Mahajan, A, and Walworth, M (2001). "3D position sensing using the differences in the time-of-flights from a wave source to various receivers", *Robotics and Automation, IEEE Transactions on*, Volume 17, Issue 1, Feb., pp 91-94

Madsen, K, Nielsen, HB, and Tingleff, O (2004). *Methods for Non-Linear Least Squares problems*, 2nd Edition, Informatics and Mathematical Modeling, Technical University of Denmark, April.

Mazl, R, and Preucil, L (2003). "Sensor data fusion for inertial navigation of trains in GPS-dark areas", *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*. 9-11 June, pp 345-350.

McCarthy, M, Duff, P, Muller, HL, and Randell, C (2006). "Accessible Ultrasonic Positioning", *University of Bristol Published by the IEEE CS and IEEE ComSoc*, pp 1536-1268.

Misra, P, Enge, P (2001). "*Global positioning system: signals, measurements, performance*". Ganga-Jumana Press, Lincoln, MA

Paperno, E, Sasada, I, and Leonovich, E (2001), "A new method for magnetic position and orientation tracking," *IEEE Trans. Magn.*, vol. 37, pp. 1938–1940, July.

Paul, RP, Sayers, CP, and Stein, MR (1993). "The theory of teleprogramming," Journal of the Robotics Society of Japan, vol. 11, no. 6, pp. 14–19, September.

Raab, FH, Blood, EB, Steiner, TO, and Jones, HR (1979). "Magnetic position and orientation tracking system," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 5, pp. 709–717.

Sayers, CP, Paul, RP, Catipovic, J, Whitcomb, LL, and Yoerger, D (1998). "Teleprogramming for subsea teleoperation using acoustic communication", *IEEE Journal of Oceanic Engineering*, vol. 23, no. 1, pp. 60–71.

Suya You, Neumann, U, Azuma, R (1999). "Hybrid inertial and vision tracking for augmented reality registration", *Virtual Reality, 1999. Proceedings., IEEE*, 13-17 March, pp 260-267.

Ulrich, RJ (1967). "Principles of Underwater Sound for Engineers", *McGraw-Hill*, NY.

Yuh, J, and Choi, SK (1999). "Semi-Autonomous Underwater Vehicle for Intervention Mission: An AUV that does more than just swim", *Sea Technology*, (40) 10:37-42.

Yuh, J, Choi, SK, Ikehara, C, Kim, GH, McMurty, G, Ghasemi-Nejhad, M, Sarkar, N, and Sugihara, K (1998). "Design of a semi-autonomous underwater vehicle for intervention missions (SAUVIM)", *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on*, 15-17 April, pp 63-68.

Intelligent, Coordinated-Motion/Force Control (ICM/FC)

Project Leader(s): Dr. Giacomo Marani
Past Project Leader(s): Dr. Junku Yuh, Dr. Tae Won Kim, Dr. Song K. Choi, Dr. Kazuo Sugihara, Dr. Hyun Taek Choi, Mr. Michael West & Dr. Nilanjan Sarkar

The main technical development of the ICM/FC group is described in the following sections: Manipulator Control and Test Platform, Low-Level Control, Active Feedback Thruster System (AFTS), Localization and Navigation, and High-Level Control. The Manipulator Control and Test Platform is the combined sections of the previous Theoretical Modeling and Dry Test Design Set-Up. The Localization and Navigation is a separation from the Low-Level Control due to the vastness and complexity of the research area.

Manipulator Control and Test Platform (MCTP)

Project Leader(s): Dr. Giacomo Marani
Personnel: Ms. Allison Lyon, Mr. Kaikala Rosa
Past Project Leader(s): Dr. Song K. Choi, Dr. Tae Won Kim, Dr. Junku Yuh & Dr. Nilanjan Sarkar
Past Personnel: Mr. Tommaso Bozzo, Mr. Gang Cheng, Ms. Jing Nie, Mr. Mike Kobayakawa, Mr. Mark Fujita, Dr. Gyoung H. Kim, Mr. Tarun Podder, Mr. Jin Hyun Kim, Mr. Jong Ho Eun, Ms. Stacy L. Dees & Mr. Jangwon Lee

Objectives

During the Phase II of SAUVIM, one of the most important objectives for the manipulation platform was the first ocean test of the system. In order to achieve the above objective, extensive product engineering works have been necessary, other than several further developments of the hardware/software control system.
The final objectives included the following:

- Further development of theoretical solutions for the arm control algorithm with extensive lab testing in order to verify the task-space controller performances.
- Development of a programming environment for manipulators which include a low level software-emulated execution CPU, a high-level programming language and a program compiler.
- Development and testing of an ultrasonic-based tracking system for target localization.
- Development of an extended subset of routines for the arm programming environment, which include a set of calibration procedures for the joint offsets and the auto-calibration of the external position sensors.
- Development of the arm parking procedures.
- Integration of the manipulator on the vehicle

The final step, after the above developments, was the first underwater manipulation experiment.
These objectives have been successfully achieved, with good performances and stability. In particular, the theoretical solutions developed for prevent singularities showed an excellent performance and were published in several journal and conferences. Details on the overall development have been described on the previous report (Phase II-C and III-A).
The Phase III-B's objectives for the manipulation platform have been focused on the development of target detection algorithms, based on camera and motion tracker. Details on the target detection developments have been reported in the AORD section of this report.
Here, we shall discuss on further support control procedure used for tasks like calibration.
In fact, one of the most important objectives for the manipulation platform was the ability to calibrate the imaging system on demand and in the water and to identify and track a target. In order to achieve the above objective, extensive software and product engineering works have been necessary.

The final objectives include the following:

- Development and testing of a visual-based tracking system for close range target localization
- Further development of theoretical solutions for the arm control algorithm with extensive lab testing in order to verify the task-space controller performances.
- Development of an extended subset of routines for the arm programming environment which include a set of calibration procedures for the auto-calibration of the camera
- Integration of the manipulator and camera system on the vehicle

The final step, after the above developments, was the underwater target following experiment.
With the system mature for testing, the Phase III-B has seen also an intensive testing in order to validate and improve the target detection procedures.

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

Summary

1. Camera
   a. Development of software for underwater camera system
   b. Development of auto-calibration procedure for underwater camera system using joint position and a calibration image
   c. Test of calibration with stationary target
   - Study of the absolute position and orientation errors of the end-effector with respect to the stationary target.

2. Camera Target Tracker
   a. Improved target acquisition and tracking algorithms with calibrated imagery
   b. Test of autonomous task with moving target
   c. Study on the absolute position and orientation errors of the end-effector with respect to the moving target

3. Integration of the manipulator with the vehicle
   - Development of the arm calibration configuration
   - Electronic system optimization for a better use of the limited number of underwater interconnections.

4. Underwater manipulation test
   - Target following experiment

Camera

The primary purpose of an intervention AUV is to perform underwater intervention missions with limited or no human assistance. The camera tracking system is used for short range manipulation of and recording of the AUVs immediate surroundings. One of the main issues regarding a camera based tracking system is overcoming the inherent defects in the camera lens along with the added distortions from the water and accurately relating the camera frame to the AUV. Autonomous manipulation capability is crucial, especially for intervention operations in a deep ocean where the low bandwidth and significant time delay are inherent in acoustic underwater communication.

The camera is attached to the sixth link of the manipulator arm. It is mounted in a fixed position so pan/tilt operations are executed through the manipulator's motion. The camera is made by Delta Vision and is rated for a depth of 4000 m. It features a wide angle lens ideal for underwater conditions. The biggest obstacle for underwater camera systems is the low lighting levels at large depths. This camera includes high light sensitivity (0.2 lux) enabling it to work reliably under low light conditions. It has a resolution of 640x420 pixels and a fixed focus ranging from 1 inch to infinity. [2]

The image processing and computer vision algorithms are performed on an INTEL Pentium-M based processor. Computations have been optimized in order to perform real time operations at a rate of 10 Hz. This speed is adequate for targets to be followed accurately through a sequence of images. The camera is calibrated prior to target detection whereby focal length, principal point, and distortion matrices are determined. [1]

In order to perform autonomous calibration additional software had to be written. The software was modified to include a variety of programmable modes. These modes are used to indicate to the software what function to perform. Currently there are modes for target tracking, calibrating the intrinsic parameters of the camera, and the transfer function from the camera frame to the manipulator. Although there are currently only 3 modes, it is now possible to quickly and easily add a variety of different functions to the AUV. These modes can either be triggered manually or programmatically.

In order to determine the intrinsic parameters of the camera such as focal length, principle point, and distortion matrices, a series of calibration images are obtained. Cameras usually exhibit significant lens distortion, especially radial distortion [6]. It is extremely important that these distortions are known and compensated for so subsequent target locating calculations are accurate.

The calibration image used is a chessboard with 1 inch squares. This image was chosen because it is the image that OpenCV, an image processing library, expects. By using OpenCV calibration calculations are optimized. OpenCV utilizes the following algorithm to calculate camera parameters:

1. Find homography for all points on a series of images
2. Initialize intrinsic parameters; distortion set to 0
3. Find extrinsic parameters for each image of pattern

4. Make main optimization by minimizing error of projection points with all parameters

Lens distortion is described by 4 coefficients, 2 radial and 2 tangential. These coefficients are calculated by OpenCV using the following equations

$$\tilde{x} = x + x[k_1 r^2 + k_2 r^4] + [2p_1 xy + p_2(r^2 + 2x^2)] \quad (1)$$

$$\tilde{y} = y + y[k_1 r^2 + k_2 r^4] + [2p_1 xy + p_2(r^2 + 2y^2)] \quad (2)$$

Where k_1 and k_2 are radial distortion coefficients, p_1 and p_2 are tangential distortion coefficients, x and y are ideal physical coordinates, $\tilde{x}$ and $\tilde{y}$ are real physical coordinates, and $r^2 = x^2 + y^2$. [6]

The general algorithm used to obtain the distortion matrices is as follows:

1. Set mode to "camera calibration"
2. Go to first position
3. Capture a number of frames until the chessboard pattern is detected
4. If the chessboard pattern is not detected, nudge the camera
5. Else move camera to the next position and repeat 3 until done

By adding in a small random number to the joints in the case a chessboard is not detected, situations where lighting or other small obstructions do not prevent the AUV from obtaining calibration data.

After the camera lens has been properly calibrated, the transformation matrix between the camera and the robotic arm can be properly attained. The location of the target in the camera means nothing without an accurate relationship between the camera and the AUV.

The general algorithm used to obtain the transformation matrix is as follows:

1. Set mode to "camera-arm transformation matrix calibration"
2. Go to first position
3. Capture a series of frames
4. Calculate the position of the chessboard
5. Move along an axis, repeat 3
6. Rotate about origin
7. Calculate transformation matrix

The transformation matrix T_t^0 of the target frame $< t >$ with respect to the base frame $< 0 >$ is given by (3)

$$T_t^0 = T_c^0 x^c \qquad (3)$$

where T_c^0 is the transformation matrix of the camera frame $< c >$ with respect to the base frame $< 0 >$, and x^c is the position of the target in the camera frame $< c >$

The placement of the camera T_c^0 is calculated by an automated algorithm. The transformation matrix is composed of two parts, a rotation segment and a translation segment given by (4)

$$T = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad (4)$$

where R is the 3x3 rotation matrix where each element is a rotation about a respective axis given by (5)

$$R = \begin{bmatrix} r_{xx} & r_{xy} & r_{zx} \\ r_{xy} & r_{yy} & r_{zy} \\ r_{xz} & r_{yz} & r_{zz} \end{bmatrix} \quad (5)$$

and P is the 3x1 position matrix given by (6). [4]

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (6)$$

Camera placement is determined through calibration by moving the manipulator arm along each axis and rotating about each axis to produce the corresponding rotation and position matrices. These matrices are given by equations (7) and (8) respectively.

$$^c_0R = \begin{bmatrix} \frac{\Delta x_1^c}{\Delta d_x} \middle| \frac{\Delta x_2^c}{\Delta d_y} \middle| \frac{\Delta x_3^c}{\Delta d_z} \end{bmatrix} \quad (7)$$

$$P_c^0 = ((R_1 - R_2) * (R_2 R_6^c m_2 - R_1 R_6^c m_1))^{-1} \quad (8)$$

Here Δx_1^c is the 3x1 matrix representing the change in the location of an object in the camera frame after the movement along one axis, x. This vector is then normalized by the distance traversed along the translated axis. The camera is then moved along the remaining axis and similar calculations are completed. R_1 and R_2 are the rotation matrices of the arm before and after a rotation around the x axis and m_1 and m_2 are the corresponding locations observed in the camera frame.

To determine the position of the camera in the base frame, a rotation of a known amount, θ, about each axis is performed. Here

$$\Delta P_x^0 = {}^0_cR x^c \quad (9)$$

Where ${}^0_cR$ is the matrix previously calculated and x^c is the position of an object in the camera frame.

The determination of the placement of the camera needs to be performed only once. It is a constant matrix corresponding to the physical position of the camera and will not change unless the camera is disturbed.

In order to test the accuracy of these calculations the robotic arm and attached camera attempted to locate a stationary target. The target used is the same target that is used in

under water experiments. It is a set of 2 spheres of a known size and a known distance apart pictured below

Figure MCTP-1. The targetinour experiments.

The location of the target relative to the camera and then to the base of the robotic arm is measured and compared to the location the camera thinks it is located. Any errors in the calculation in the location of the target or calibration of the camera would have resulted in a false position.

Camera Target Tracker

A properly calibrated camera greatly improves the ability of the robotic arm and vehicle to both obtain the target and to track it. Calibrating the camera for lens distortions provides an increased ability to locate the spherical targets because the spheres appeared more true to their form. It also improves the ability of the arm and vehicle to track it since the position of the target is directly determined from the radius of the circle and location on the image plane.

Extraction of the target is completed through a sequence of the following image processing techniques

- Image distortion compensation
- Image filtering
- Edge extraction using a Canny filter applied to the color image and using the color contrast gradient
- Circle extraction using the line segments found
- Circles smaller than some threshold are immediately thrown out

This combination of algorithms gives the best performance and robustness with respect to false or missed detections in an underwater environment. [1]

After the circles are located, the location of the target with respect to the camera x^c is calculated using the radius and the center of the circles extracted from the image.

$$Z = SPHERE\_RADIUS * m\_f/r \quad (10)$$
$$X = Z * (u - m\_uc)/m\_fx \quad (11)$$
$$Y = Z * (m\_imgH - v - m\_vc)/m\_fy \quad (12)$$

The camera frame is located at the center of the frame of the image with the Z direction the distance from the camera. In these equations X, Y, and Z is the position of the target in the X, Y, and Z axis respectfully. SPHERE_RADIUS is the known radius of one of the dipoles of the target. R is the radius of the circle detected through image processing. The terms u and v are the pixel locations of the center of the sphere where pixel (0, 0) is the pixel in the upper left corner of the image. The quantities m_f, m_fx, and m_fy are the focal lengths obtained from the camera calibration procedure and are given in pixels. Finally, m_imgH is the height of the image in pixels. [3]

By comparing the 3D position of the circles to one another, potential targets are extracted. Only these circles go on to the location estimation step. This step cuts down the quantity of false detections because it is very unlikely that two random circles falsely extracted from the image will coincide with the proper 3D distance between the two target spheres.

To test the accuracy of this procedure a test with a moving target was performed. With the AUV in "tracking mode" the camera attempted to locate the target and track it. By moving the target and observing the response of the robotic arm it was clear that the arm was following the target and adjusting its angle for different target orientations.

Underwater Manipulation Test

The goal for the underwater manipulation test was to track a dipole underwater object with the camera. This demo is a simplified version of the original SAUVIM test in shallow water. It included all components of the SAUVIM project such as vehicle navigation and control, landing/station keeping, robot control, and image processing, acoustic image. The following steps were performed in the last site visit in June, 2008:

1. Deploy SAUVIM (with wireless buoy and tether) on the surface.
2. Use SAUVIM to scan the test site to make a rough seafloor map with down facing Imagenex sonar, TCM2 heading sensor, and GPS.
3. SAUVIM returns to the start location in the water to show its underwater navigation performance
4. Move SAUVIM to the area of the target
5. SAUVIM initiates an autonomous scan for the target
6. Once target is acquired SAUVIM begins station keeping
7. SAUVIM deploys the robot manipulator.
8. SAUVIM grabs a hooking tool from a holder on the vehicle.

9. SAUVIM scans the front area of the vehicle with a camera on the robot, until the target is detected.
10. SAUVIM measures the relative distance and angle from detected target image.
11. SAUVIM continues to station keep in front of the target, compensating for currents and target movements.
12. SAUVIM sends GPS data of its location to the ground station for vehicle retrieval.

References

[1] Marani, G., Yuh, J. and Choi, S. K. (2007). "Experimental Study on Autonomous Manipulation for Underwater Intervention Vehicles," " in *Proc., IEEE Int. Conf. on Robotics and Automation*, pp. 1973–1978.

[2] Marani, G., Yuh, J. and Choi, S. K. (2006). "Autonomous manipulation for an intervention AUV," in *Guidance and Control of Unmanned Marine Vehicles*, G. Roberts and B. Sutton, Eds. IEE's Control Engineering Series, pp. 217–237.

[3] H. Mahesh, J. Yuh, and R. Lakshmi, "A coordinated control of an underwater vehicle and robot manipulator," *J. Robot. Syst.*, vol. 8, no. 3, pp. 339–370, 1991.

[4] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Reading, MA: Addison-Wesley, 1989.

[5] B.K.P. Hron., Computer Vision, MIT Press, 1986

[6] Intel, Open Source Computer Vision Reference Library, http://www.developer.intel.com, 2001

Low-Level Control (LLC)

Project Leader(s): Dr. Giacomo Marani
Personnel: Dr. Giacomo Marani
Past Project Leader(s): Dr. Side Zhao, Dr. Junku Yuh, Dr. Song K. Choi, Dr. Tae Won Kim & Dr. Hyun Taek Choi
Past Personnel: Ms. Jing Nie, Mr. Eric Kardash & Mr. Michael West

Objectives

- To design an advanced vehicle control for navigation and hovering, and coordinated motion/force control of the vehicle and manipulator during the intervention mode.
- To develop hybrid controllers that is robust to system uncertainties as well as external disturbances of the AUV dynamics.

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

- Development and implementation 6 DOF SAUVIM Position Controller
- Development of SAUVIM dynamics model;
- Development of time-optimal trajectory algorithm.

The primary purpose of an autonomous manipulation system is to perform intervention tasks with a limited exchange of information between the manipulator and the human supervisor. The information passed to the main control system is often only a high level decision command, and the controller must be capable of following the decision command by providing reliable control references to the actuators.

The main issue in designing and implementing a control system for autonomous vehicles is ensuring a reliable behavior, which means also avoiding singularities, collisions, system instabilities and unwanted motions while performing the required task is theoretically executable.

The control system for an intervention vehicle must also address some general manipulation issues, such as being task-space oriented, with task priority assignments and dynamic priority changes.

The third layer of the main control diagram of Figure LLC-1 is the Medium Level Controller of the system and it is the layer where the above issues are addressed. This chapter describes in details the approach adopted in order to solve the kinematical problems inherent a control system for and intervention AUV.

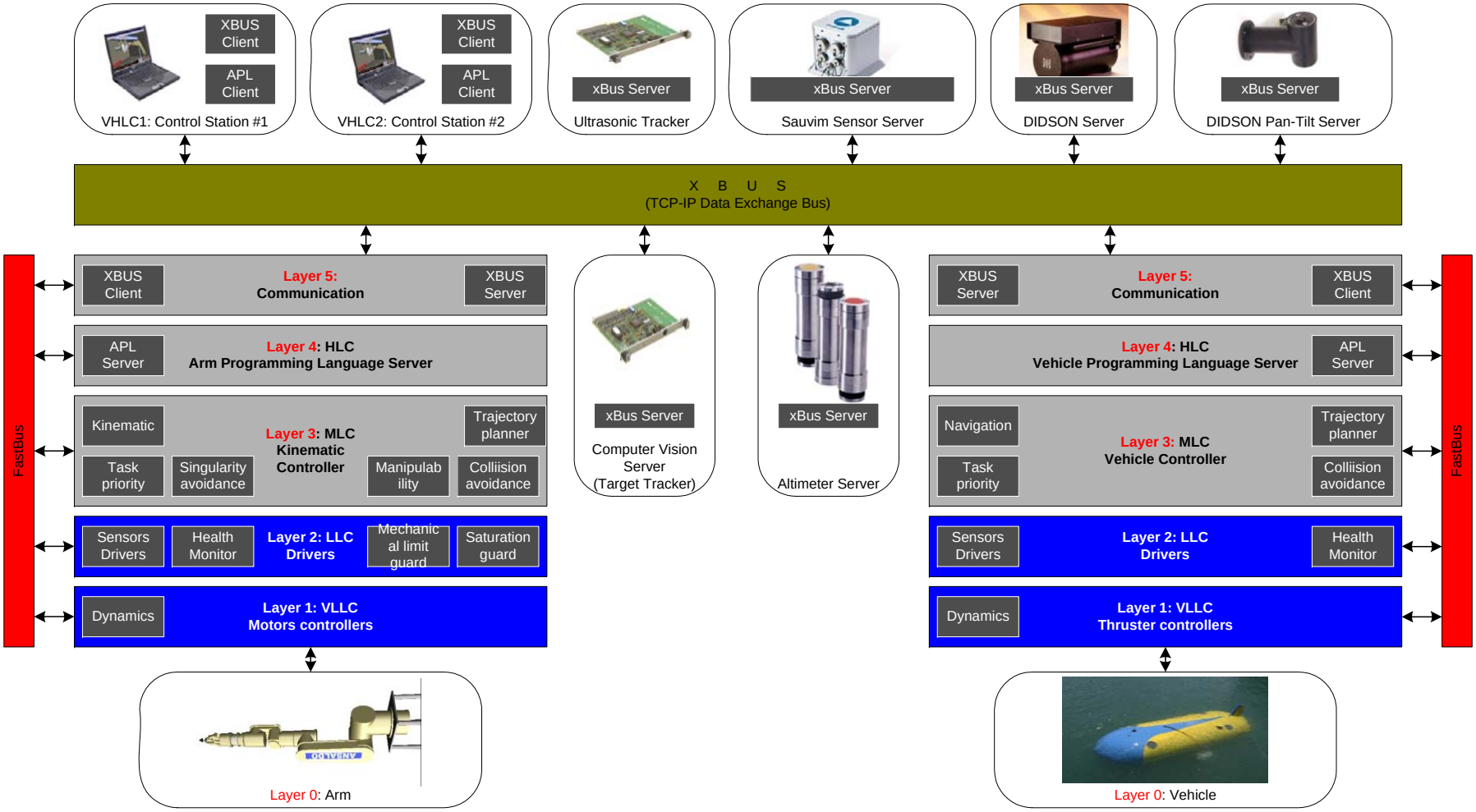


Figure LLC-1. The new SAUVIM control diagram.

Generation of the velocity reference

In this chapter we are not considering the problem inherent the dynamic control of the vehicle. In our treatment, the vehicle together with its dynamic controller is considered as a separate subsystem, as shown in Figure LLC-2. The function of the kinematic controller described here is to generate an appropriate velocity profile for the velocity controller.

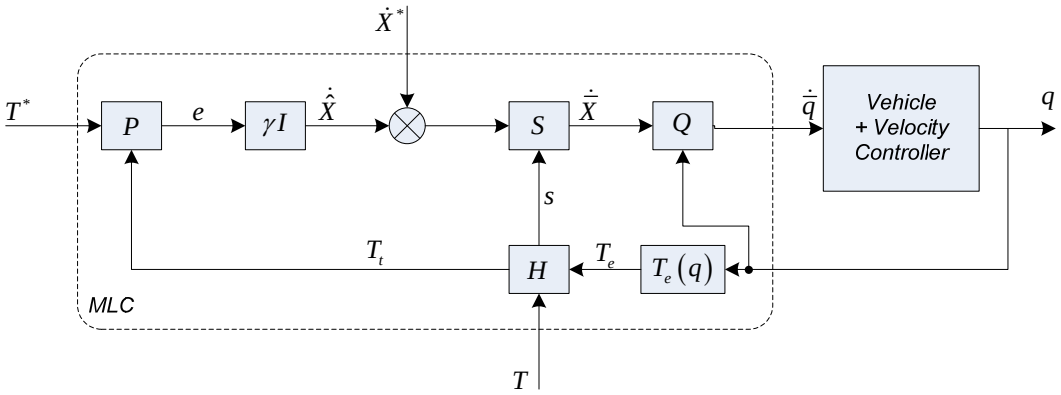


Figure LLC-2. The SAUVIM control scheme.

In the control scheme of Figure LLC-2, the block named "Vehicle + Velocity Controller" represents the physical vehicle equipped with its velocity controller. The overall block can be seen as a compact one, receiving the vector of the reference joint velocities as input, and giving the vector of the corresponding vehicle positions as output.

Closing the feedback loop

Let's consider a schematic representation of a the SAUVIM vehicle and its workspace as in Figure LLC-3. Here, ${}^0_S T$ is the transformation matrix of the SAUVIM frame $< S >$ with respect to the base frame $< 0 >$, while ${}^0_T T$, generally time varying, is the transformation matrix of the reference (target) frame $< T >$ with respect to the base frame $< 0 >$. The reference frame $< T >$ is usually computed in order to place the target in the manipulator workspace.
The general goal is to track the reference frame ${}^0_T T$ by the SAUVIM frame $< S >$. At this aim, the global error *e* is automatically defined by the vector

$$e \doteq \left[r_{gt}, \rho_{gt} \right]^T$$

where vectors r_{gt} and ρ_{gt} (both projected on the base frame $< 0 >$) represent the distance and the misalignment (equivalent rotation vector) of the reference frame $< T >$ with respect to $< S >$. The objective of the control scheme is to make the global error *e* asymptotically converging toward zero or, alternatively, asymptotically confined within acceptable norm bounds. This goal could be achieved with the closed loop scheme shown in Figure LLC-2.

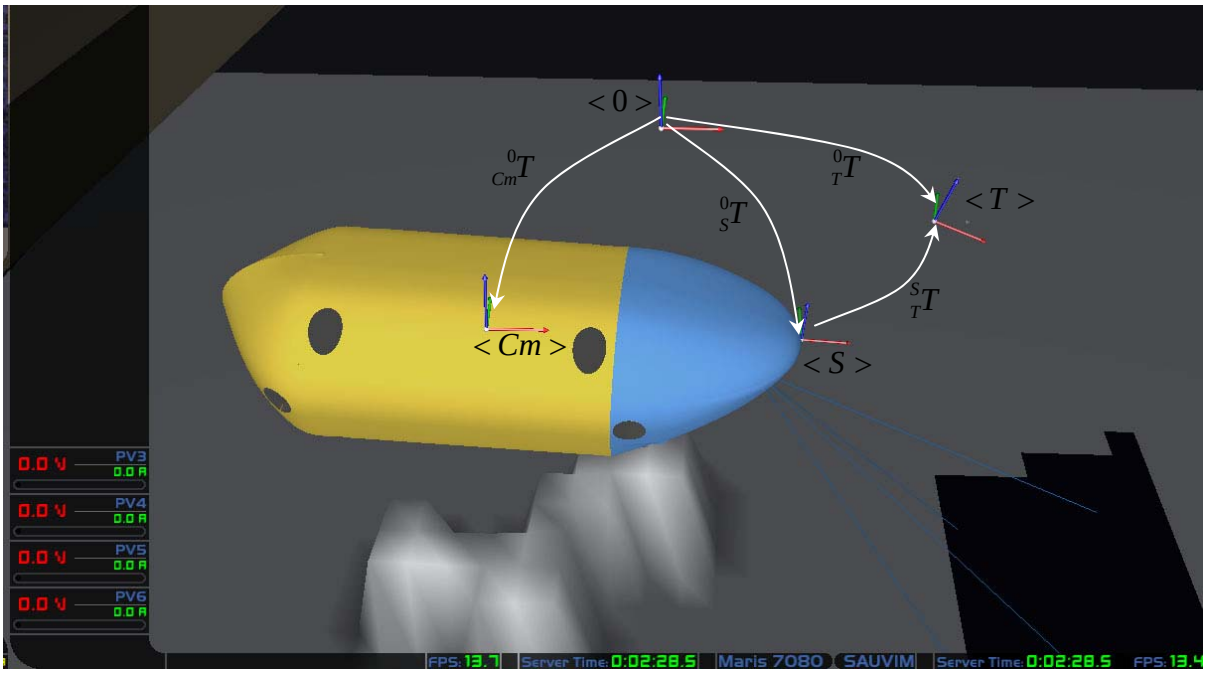


Figure LLC-3. The SAUVIM Vehicle and its frames.

Medium Level Control Loop

The remaining part of the control system represents the Medium Level Control (MLC) loop of the vehicle. The center of mass generalized velocity reference $\dot{\bar{q}}$ is appropriately generated as real-time outputs, such that the global error e converges toward the specified bounds. The reference transformation matrix $^{0}_{T}T$ is compared with the actual SAUVIM frame $< S >$ via the processing block P, which is used for evaluating the global error e in real time by solving, for the rotational error part ρ_{gt} only, the well known "versor lemma" equations, given by:

$$\begin{cases} i_t \wedge i^* + j_t \wedge j^* + k_t \wedge k^* = \dfrac{1}{2} z \sin\theta \\ i_t^T \cdot i^* + j_t^T \cdot j^* + k_t^T \cdot k^* = 1 + \cos\theta \end{cases}$$

with $R_t \doteq [i_t, j_t, k_t]$, $R^* \doteq [i^*, j^*, k^*]$ the rotation matrices contained inside the transformation matrices $^{0}_{S}T$ and $^{0}_{T}T$ respectively, while $\rho \doteq z\theta$ with z a unitary vector and θ an angular quantity. The notation $a \wedge b$ is used for indicating the cross-product of two generic three dimensional vectors a and b.

The linear part r_{gt} of the global error is easily obtained as difference between the first three elements of the last columns of ${}^{0}_{T}T$ and those of ${}^{0}_{S}T$. The global error e is then multiplied by a suitable gain matrix γI. The result is the generalized Cartesian velocity $\dot{\hat{X}} \doteq \left[\hat{\omega}, \hat{v}\right]^T \in \Re^6$ (projected on $< 0 >$), where $\hat{\omega} \in \Re^3$ and $\hat{v} \in \Re^3$ are the angular and linear velocity, respectively, which are assigned to the SAUVIM frame $< S >$ such that e converges within the specified bounds. At this stage, the additional Cartesian velocity input $\dot{X}^*$ allows a direct control of the SAUVIM velocity.

The SAUVIM generalized velocity control signal $\dot{\bar{X}}$ is transformed into a corresponding center of mass velocity vector $\dot{\bar{q}}$ by the functional block Q.

Active Feedback Thruster System (AFTS)

Project Leader(s): Mr. Aaron Hanai
Personnel: Mr. Kaikala Rosa, Mr. Christopher McLeod

Objectives

Since one of the primary goals for the vehicle is underwater manipulation, the thruster subsystem must be accurate enough to maintain robust hovering of the vehicle. This has required experimental analysis and tuning of the hardware and engineering design into the performance of the software feedback control scheme. The objectives of this system include:

- An energy efficient distribution of forces among the 8 vehicle thrusters using an analytical approach (as opposed to heuristics).
- A closed-loop thruster control design based on feedback from the motor controllers.
- A software supervisor to prevent errors when the reference thrust exceeds the physical limits of the hardware due to voltage sag in the source batteries.
- A model-based thrust estimator that is robust to unfavorable water conditions in which cavitation may occur.

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

1. Thruster force allocation:
   - Definition of a variable thruster configuration matrix mapping between the thruster forces and the body-fixed vehicle forces/torques
   - Solution of the thruster configuration matrix via weighted pseudoinverse

2. Saturation Guard:
   - Separation and isolation of the linear and angular thrust errors
   - Modeling of thrust loss due to battery voltage sag

3. Thruster modeling:
   - Experimental analysis of the functional relationships between thruster input reference voltage, measured current, measured velocity, and output thrust
   - Development of model-based thrust approximation functions

4. Cavitation tolerance:
   - Experimental observation of the effects of cavitation on thruster performance
   - Development of a model-based fault tolerant thrust estimation function (robust to cavitation)
   - Development of a fault accommodating thruster system that scales the thruster configuration matrix based on the error estimation

Active Feedback Thruster System

The new active feedback thruster system is comprised of the components displayed in Figure AFTS-1. Hardware elements are outlined in blue, and software in black. Following the diagram, the sensors feed kinematic information (position, velocity, acceleration) about the vehicle to the navigation controller. The controller develops a six-term body-fixed reference force vector τ_{ref} and sends it to the thrust management block in exchange for an eight-term estimated thruster force vector T_{est}. The thrust management block collects feedback information (electrical current and propeller shaft velocity I_m and U_m respectively) from the motor controllers and combines the information with the input reference τ_{ref} to generate control voltages V_m for the motor controllers. The motor controllers operate the thrusters in current-mode, in which the controllers vary the output velocity in order to maintain the current to the thrusters, relative to the control voltage input.

Figure AFTS-1. Block Diagram Overview

Thruster Force Allocation

The vector τ is the body-fixed input of the vehicle equation of motion

$$A(q)\dot{p} + B(q,p)p + C(q) = \tau \qquad \text{(AFTS-1)}$$

where

$$q = \begin{bmatrix} roll \\ pitch \\ yaw \\ x \\ y \\ z \end{bmatrix} \quad p = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ v_x \\ v_y \\ v_z \end{bmatrix} \quad \tau = \begin{bmatrix} M_x \\ M_y \\ M_z \\ F_x \\ F_y \\ F_z \end{bmatrix}. \qquad \text{(AFTS-2)}$$

The navigation controller solves this equation of motion in order to generate a reference vector τ_{ref} from the values $q, p, \dot{p}$ from the sensors. This vector must be transformed to a thruster reference vector $T = [T_1 \ \cdots \ T_8]$ according to the geometry in Figure AFTS-2.

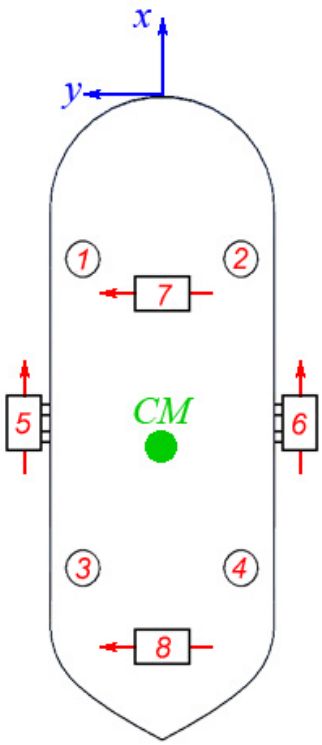


Figure AFTS-2. SAUVIM Thruster Geometry

Defining the thruster positions and orientations p_i and r_i respectively, as well as the vehicle center of mass $C = [C_x, C_y, C_z]$, then by geometry, $\tau = KT$ according to the transformation matrix

$$K = \begin{bmatrix} (p_1 - C) \times r_1 & \cdots & (p_8 - C) \times r_8 \\ r_1 & \cdots & r_8 \end{bmatrix} \tag{AFTS-3}$$

At this point the equation can be inverted such that

$$T = K^{\#}\tau\,, \tag{AFTS-4}$$

according to the generalized inverse

$$K^{\#} = W^{-1}K^T\left(KW^{-1}K^T\right)^{-1}, \tag{AFTS-5}$$

which minimizes the error norm $\|\tau - KT\|$. The weight matrix W is such that

$$W^{-1} = \begin{bmatrix} w_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_8 \end{bmatrix} \tag{AFTS-6}$$

where the coefficients $0 \le w_i \le 1$ reflect the thruster reliabilities. In this case $(w_i = 1)$ represents complete functionality, whereas $(w_i = 0)$ represents complete thruster failure. Since the generalized inverse of K yields the minimum-energy particular solution, more energy can be utilized in order to avoid the thruster dead zones. This can be accomplished by adding a homogeneous solution from the nullspace of K. Given $\tau = KT$, a nonzero homogeneous solution $T_{null} \neq 0$ is such that $\tau(T_{null}) = 0$. Since the projection onto the nullspace of K is given by $I - K^{\#}K$, for arbitrary vector z,

$$T_{null} = \left(I - K^{\#}K\right)z\,. \tag{AFTS-7}$$

Since K is a 6-by-8 matrix, if it has full rank of 6, then its nullity is 2. That is, the rank of the nullspace is only 2. Hence, there are only 2 scale factors that affect the nullspace. Physically, it is one for the vertical motions and one for the horizontal motions. Therefore, the vector z can be expressed in the form

$$z = \begin{bmatrix} V_{scale} & 0 & 0 & 0 & H_{scale} & 0 & 0 & 0 \end{bmatrix}^T \tag{AFTS-8}$$

Saturation Guard

If the reference thrust for a particular thruster exceeds its physical limit, the resulting error may propagate to both the linear and angular vehicle velocity and acceleration. This phenomenon is illustrated in a simple two-dimensional fashion in Figure AFTS-3. In this diagram, the blue arrows represent the physical thrust limits of the two thrusters. The dashed red arrow is the resultant output of the vector components in solid red. The first two rows of Figure AFTS-3 demonstrates how a reference thrust in which one of the vector components exceeds its physical limit, will have errors in both magnitude and direction. Since each thruster limit $T_{i,lim}$ can be measured and therefore known, a thruster can be classified as saturated if the term $S_i = \left(T_{i,ref} / T_{i,lim}\right) > 1$. A saturation guard can be implemented, such that if any one or more of the thrusters are in saturation, then scale the entire vector T by the factor $1/\left(S_i\right)_{max}$.

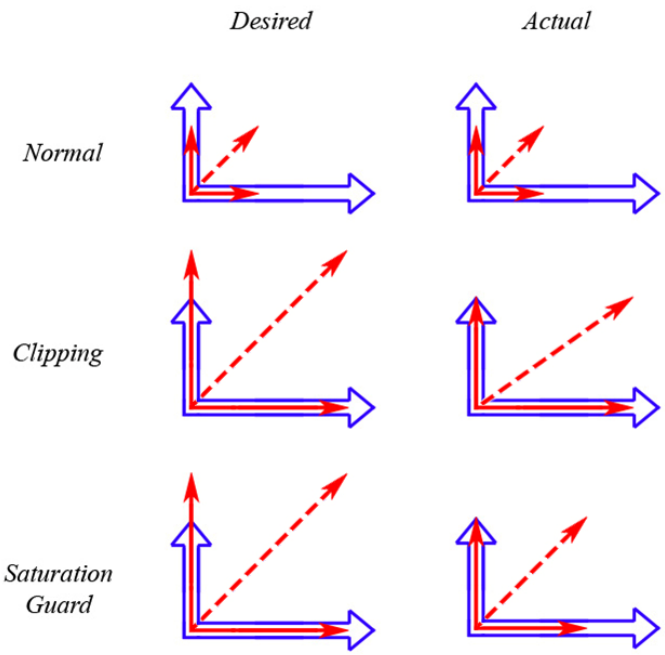


Figure AFTS-3. Saturation Guard Illustrative Example

In practice, the physical limits of the thrusters are not constant because they are a function of the source battery voltage, which in turn sags as it loses charge. This loss of thrust is illustrated in the measured data plotted in Figure AFTS-4. A model was derived to estimate the thrust limits (for use by the saturation guard) as a function of battery voltage, and is shown in Figure AFTS-5.

Thruster Modeling

To develop the thruster models and thrust approximations, experiments were performed to measure the relationship between the control input voltage, feedback current, feedback velocity, and thrust (measured via load cell). These relative measurements are displayed versus time in Figure AFTS-6. In the process of the experimentation, the motor controllers were all tuned to the same baseline settings of gain, zero offset, and current limits. Furthermore, due to the upgrade of the battery system from lead-acid to NiMH technology, the gain and current limits

could be increased for additional performance. The resulting increase in output thrust is up approximately 50% compared to a year ago, and is displayed in Figure AFTS-7. Because the current, velocity, and thrust data were collected simultaneously with a common time stamp, different permutations of these three data sets could be analyzed in order to develop mappings from one to another, based on the requirements of the thrust management block.

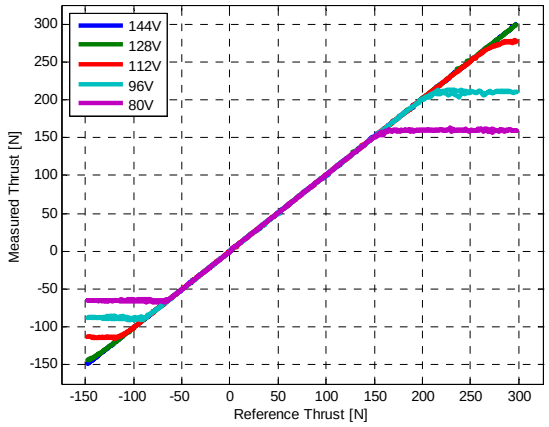


Figure AFTS-4. Measured versus reference thrust as a function of source battery voltage

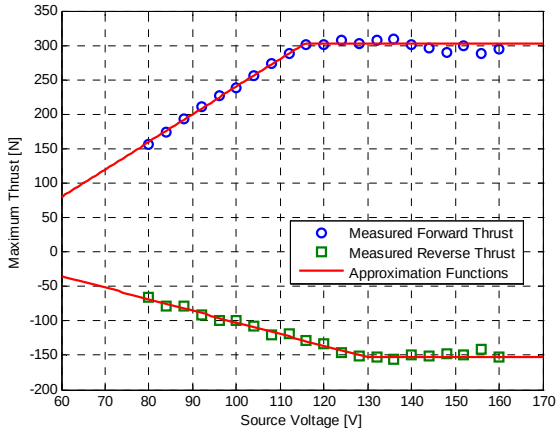


Figure AFTS-5. Maximum thrust versus source battery voltage

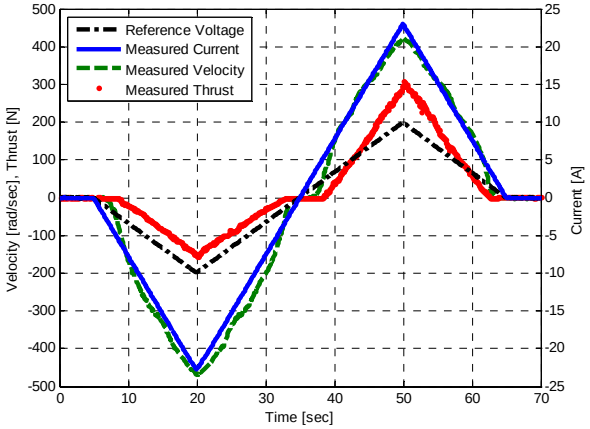


Figure AFTS-6. Voltage, current, velocity, and thrust versus time

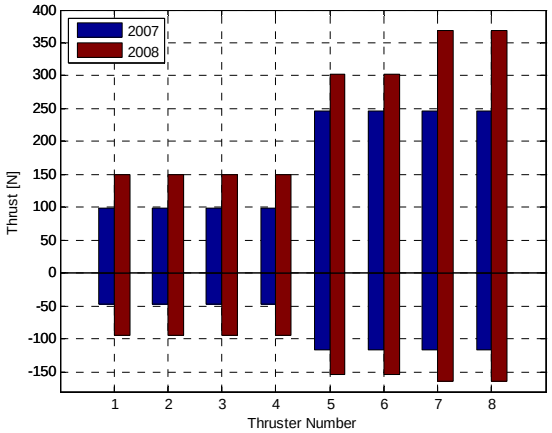


Figure AFTS-7. Thrust output comparison

Recall that the navigation controller sends a reference body-fixed τ to the thrust management block, which then transforms it to a thrust vector $T = K^{\#}\tau$. These thrust values must then be converted to motor controller input voltages. Voltage was plotted versus thrust, and a smooth piecewise-continuous function was fit to the data as in Figure AFTS-8. The thrust to voltage data fit has the form

$$V(T)=\begin{cases} aT+bT^{1/2}+cT^{1/3}+dT^{1/4} & (T\geq 0) \\ fT+gT^{1/2}+hT^{1/3}+kT^{1/4} & (T<0) \end{cases} \quad \text{(AFTS-9)}$$

and was chosen so that the function is continuous near zero in order to mitigate potential chatter due to small oscillations around the origin.

Two independent thrust approximation functions were developed. The current to thrust function, plotted in Figure AFTS-9, was fitted as two linear approximations (forward and reverse) with a dead zone in between. The function has the following form:

$$T_{est}(I)=\begin{cases} aI+b & \left(I>\frac{-b}{a}\right) \\ cI+d & \left(I<\frac{-d}{c}\right) \\ 0 & \left(\frac{-b}{a}<I<\frac{-d}{c}\right) \end{cases} \quad \text{(AFTS-10)}$$

The velocity to thrust function, plotted in Figure AFTS-10, was fitted as two 5-degree polynomials that are continuous through the origin, as follows:

$$T_{est}(U)=\begin{cases} aU^5+bU^4+cU^3+dU^2+fU & (U\geq 0) \\ gU^5+hU^4+kU^3+mU^2+nU & (U<0) \end{cases} \quad \text{(AFTS-11)}$$

To verify the accuracy of the two thrust approximations, they were plotted alongside the measured thrust versus time in Figure AFTS-11 and demonstrated favorable results.

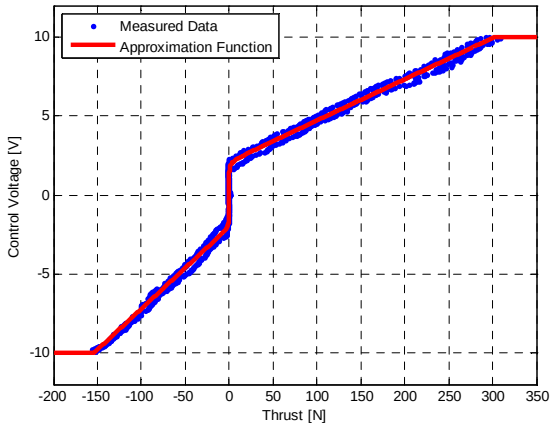


Figure AFTS-8. Voltage versus thrust

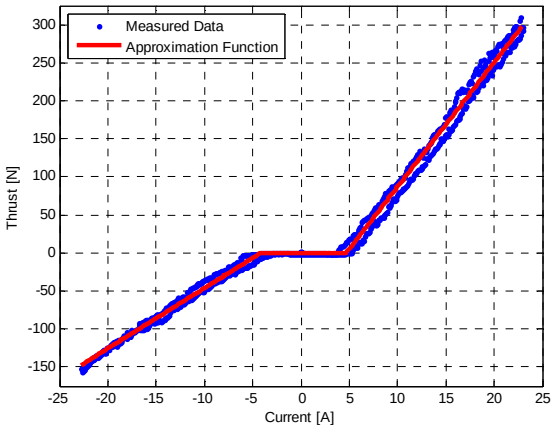


Figure AFTS-9. Thrust versus current

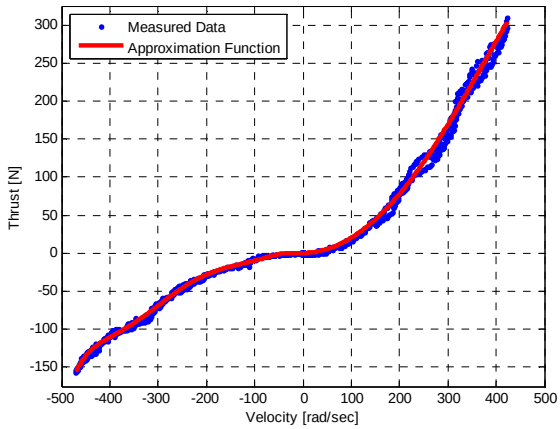


Figure AFTS-10. Thrust versus velocity

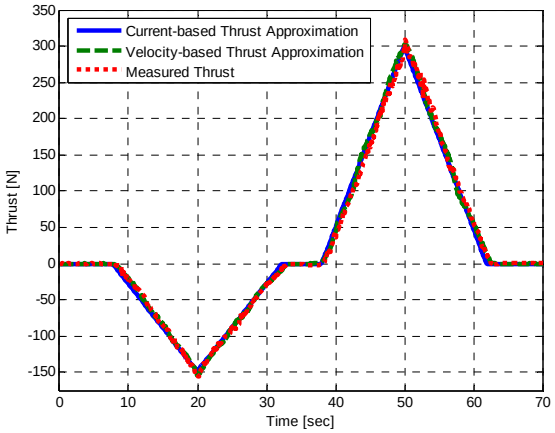


Figure AFTS-11. Thrust versus time

Cavitation

The purpose for developing two independent approximations for thrust is apparent when conditions become unfavorable. When cavitation occurs, which is common if the vehicle is at the surface, the two thrust approximations deviate from each other. In the first half of Figure AFTS-12, the water conditions are favorable, and therefore the measured thrust follows the reference, and the two thrust approximations agree with the measured value. However, in the second half of Figure AFTS-12, the thruster is positioned near the water surface so that cavitation occurs. In this case, the measured thrust no longer follows the reference, and the velocity-based thrust approximation is overestimated relative to the measured value, while the current-based approximation is also overestimated, but to a lesser extent.

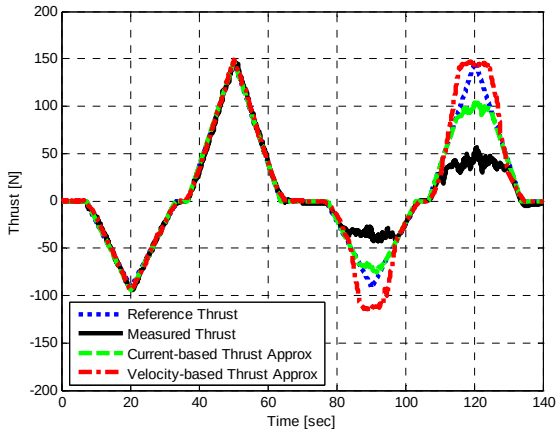


Figure AFTS-12. Thrust approximation inaccuracy (favorable conditions from 0-70 sec, cavitation occurs from 70-140 sec)

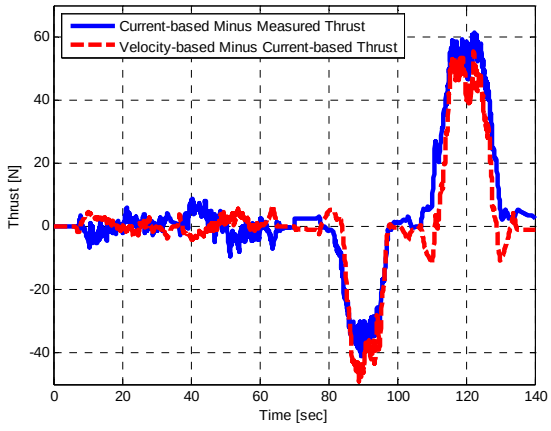


Figure AFTS-13. Thrust observation during cavitation: $T_{est}(U) - T_{est}(I) \approx T_{est}(I) - T_{meas}$

An observation was made that the difference between the two thrust approximations is about equal to the difference between the current-based approximation and the measured value. This relationship can be expressed as follows:

$$T_{est}(U) - T_{est}(I) \approx T_{est}(I) - T_{meas} \qquad \text{(AFTS-12)}$$

and is plotted in Figure AFTS-13. This relationship can be solved to develop a new thrust approximation that accounts for cavitation by considering both current and velocity as follows:

$$T_{est}(I,U) = 2 \cdot T_{est}(I) - T_{est}(U) \qquad \text{(AFTS-13)}$$

This function proved to be linear with the measured thrust, as shown in Figure AFTS-14. The experiment performed in Figure AFTS-15 is identical to that in Figure AFTS-12, except that the new thrust approximation function is able to perform satisfactorily in spite of cavitation.

Per thruster, the thrust estimate can be combined with the reference thrust to generate a weight value according to

$$W = \frac{T_{est}(I,U)}{T_{ref}}. \qquad \text{(AFTS-14)}$$

In this relationship, plotted in Figure AFTS-16, during favorable conditions, the weight is 1, and scales down as the thrust approximation deviates from the reference. The eight scalars can be combined into a vector and used in the τ to T transformation from Eq. (AFTS-4).

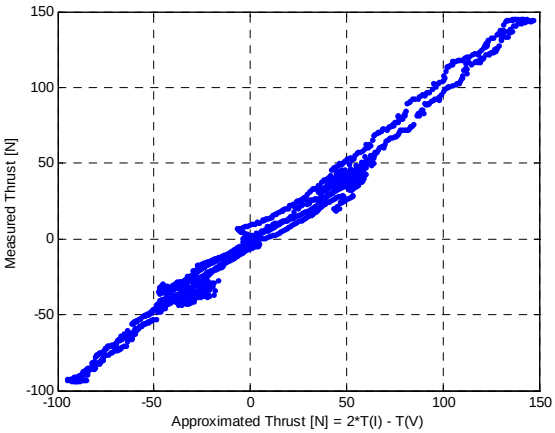


Figure AFTS-14. Measured thrust versus approximated thrust

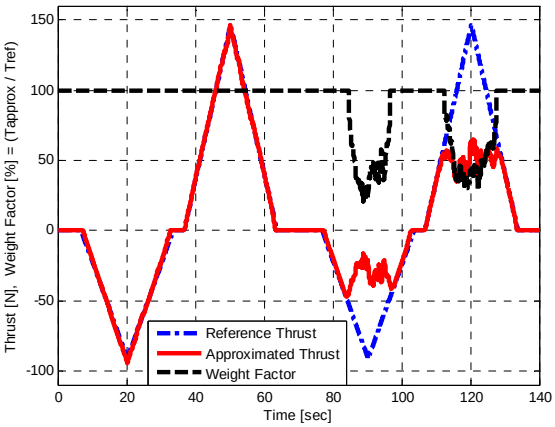


Figure AFTS-16. Thruster weights

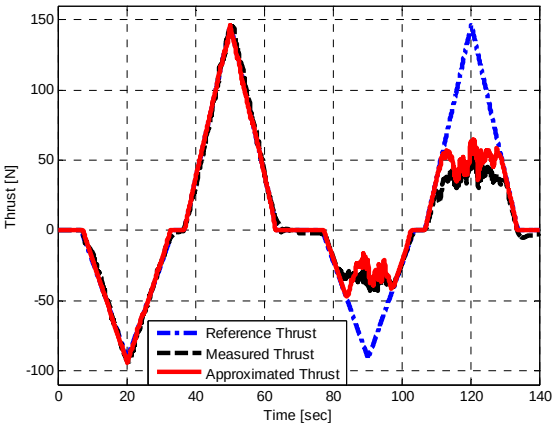


Figure AFTS-16. New thrust approximation

Localization and Navigation (LN)

Project Leader(s): Dr. Giacomo Marani
Personnel: Dr. Son-Cheol Yu, Dr. Giacomo Marani
Past Project Leader(s): Dr. Son-Cheol Yu, Dr. Junku Yuh, Dr. Tae Won Kim, Dr. Song K. Choi & Mr. Michael West
Past Personnel: Mr. Kaikala H. Rosa, Mr. Scott A. Menor, Mr. Daniel Shnidman & Mr. Mike Hall

Objectives

Global Localization of SAUVIM in all the different condition (on the surface and underwater).

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

Introduction

This technical report describes the solution adopted in order to reliably integrate the position and velocity data from all the different sensors within SAUVIM.
The presented approach uses the Extended Kalman Filter in order to correct the data according to the most reliable source.
This Extended Kalman Filter library is powerful and very simple to use, but a Kalman filter is very difficult to debug. So, it is very important to follow a procedure to be sure that everything is right (code and equations).

Data collection

SAUVIM collects data from different sensors source:

- DGPS data. The position from the DGPS sensor is absolute, with an accuracy of about a meter. This accuracy may change with the time, due to the relative motion of the satellites with respect to the earth.
- DVL data. The DVL provides accurate velocity with respect to the bottom. However, these velocities must be integrated using mainly the heading information.
- FOG (Fiber Optic Gyro). The FOG gives the heading information of the vehicle (Yaw angle) with a precise relative accuracy, but with also a slow-drifting offset which changes with the temperature.
- TCM2. The TCM2 provides the absolute rotation information of the vehicle (Roll, Pitch and Yaw with respect to the earth). However the accuracy is limited and subjected to magnetic disturbance.

All the data must be processed and transformed into the SAUVIM Generalized Position Vector, given by:

$$\mathbf{X} = \begin{bmatrix} r \\ p \\ h \\ x \\ y \\ z \end{bmatrix} \tag{2.1}$$

where r, p, and h are respectively the roll, pitch and yaw angles of the vehicle frame $< S >$ with respect to the earth frame $< 0 >$ and x, y, z are the cartesian positions with respect to the origin. Figure 1 shows the placement of the earth frame and the physical meaning of the yaw angle h. The z axis, not reported in figure 1, is directed toward the sky (the coordinate system follows the orthogonal right hand rule). Roll (r) and pitch (p) angles follow by definition.

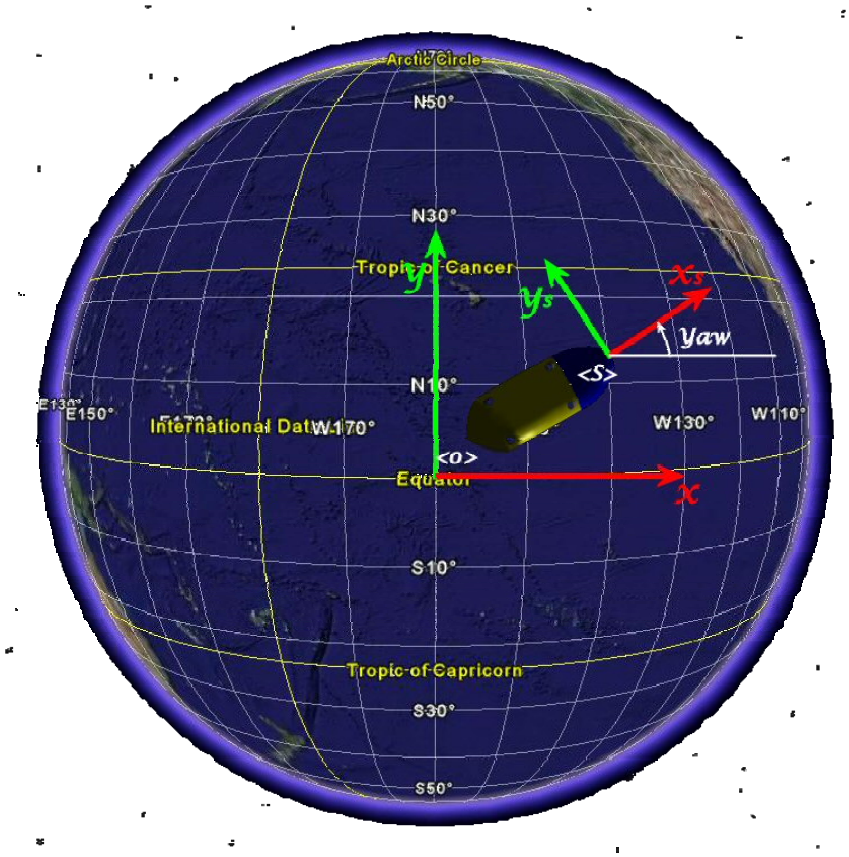


Figure 1. SAUVIM and the earth coordinates.

Every sensor provides information with respect its internal frame. For example, the TCM2 provides the roll, pitch and YAW rotation of its body with respect to the earth coordinates. Figure 2 shows its placement within the vehicle. There, the rotation matrix ${}^{S}_{T}R$ is defined so that:

$${}^{S}\mathbf{v} = {}^{S}_{T}R \cdot {}^{T}\mathbf{v} \tag{2.2}$$

where ${}^{S}\mathbf{v}$ is a generic 3D vector represented in the SAUVIM frame $< S >$ and ${}^{T}\mathbf{v}$ is the same vector projected in the frame $< TCM2 >$.
Indicating with ${}^{0}_{S}R$ the rotation matrix of SAUVIM with respect to the earth, and with ${}^{0}_{T}R$ the rotation matrix of the TCM2 with respect to the earth, we have:

$$ {}^{0}_{S}R = {}^{0}_{T}R \cdot {}^{T}_{S}R = {}^{0}_{T}R \cdot \left({}^{S}_{T}R \right)^{T} \tag{2.3} $$

Hence, in order to compute the orientation of SAUVIM with respect to the earth frame we need to identify the placement matrix ${}^{S}_{T}R$ of the TCM2.
The same must be done with all the other sensors.

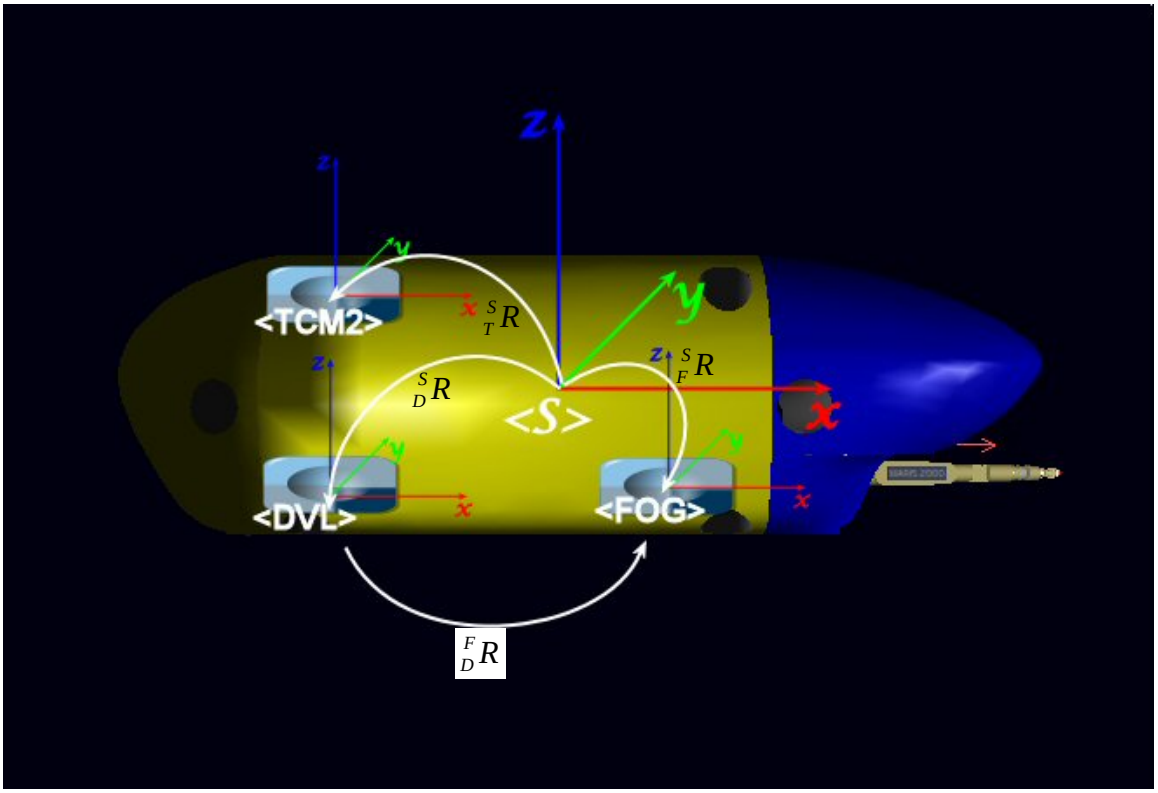


Figure 2. SAUVIM and the sensors.

In particular, since the FOG provides only the yaw angle, its rotation matrix ${}^{S}_{F}R$ is just a constant rotation around the z axis:

$$ {}^{S}_{D}R = \begin{bmatrix} \cos(\delta_{SF}) & -\sin(\delta_{SF}) & 0 \\ \sin(\delta_{SF}) & \cos(\delta_{SF}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4} $$

The Extended Kalman Filter

The uncertainties intrinsic in our data source naturally suggest the use of a Kalman filter for processing our data. As a matter of fact, our input data are:

1) The absolute cartesian position w.r.t. the earth from the GPS
2) The cartesian velocity w.r.t the bottom from the DVL
3) The heading angle relative to the starting position from the FOG
4) The absolute heading angle w.r.t. the earth from the TCM2 and the DVL

Each quantity is affected by different levels of noise.

In order to apply the Kalman filter to our problem the first thing to do is to find out the state vector describing our model for the sensor data. In general, the non-linear process function that describes the evolution of the state vector through time is:

$$\vec{x}_k = f\left(x_{k-1}, u_{k-1}, w_{k-1}\right) \tag{3.1}$$

where w is the process noise vector due to uncertainty and process modeling errors.

The relation (in general non-linear) between the state vector $\vec{x}$ and the measure vector $\vec{z}$ is:

$$\vec{z}_k = h\left(z_{k-1}, v_{k-1}\right) \tag{3.2}$$

where v is the measure noise vector.

Our problem is to estimate the best cartesian position, using al the possible data source and can be represented by the following differential equations:

$$\begin{cases} \dot{x} = k_x v_x \cos(\theta) - k_y v_y \sin(\theta) \\ \dot{y} = k_x v_x \sin(\theta) + k_y v_y \cos(\theta) \\ \theta = \theta_{FOG} + \delta_{FOG} \end{cases} \tag{3.3}$$

where x and y are the cartesian position of the vehicle w.r.t. the Earth (main) frame; v_x and v_y are the cartesian velocities measured by the DVL, in the DVL (beam?) reference frame; k_x and k_y are two corrective factors, necessary to match the scale of the GPS with the one of the DVL; θ_{FOG} is the reading from the Fiber Optical Gyro and δ_{FOG} is the offset angle of the FOG reference frame $< FOG >$ with respect to the earth frame $< 0 >$.

The GPS measures the absolute position of the vehicle w.r.t. the Earth (main) frame:

$$\begin{aligned} {}^0x &= x_{GPS} \\ {}^0y &= y_{GPS} \end{aligned} \tag{3.4}$$

Finally, the both the TCM2 sensors (one is inside the DVL) provide the absolute rotation of the vehicle w.r.t. the Earth frame:

$$\theta = \frac{\theta_{TCM\,2a} + \theta_{TCM\,2b}}{2} \tag{3.5}$$

Time and measurement update equations

Within the above scenario, we can assume the state vector being represented by the following equations:

$$\vec{x}_k = \begin{bmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \\ (k_x)_k \\ (k_y)_k \\ \theta_k \\ (\delta_{FOG})_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + T\dot{x}_{k-1} \\ (k_x)_{k-1} v_x \cos(\theta_{k-1}) - (k_y)_{k-1} v_y \sin(\theta_{k-1}) + \omega_1 \\ y_{k-1} + T\dot{y}_{k-1} \\ (k_x)_{k-1} v_x \sin(\theta_{k-1}) + (k_y)_{k-1} v_y \cos(\theta_{k-1}) + \omega_2 \\ (k_x)_{k-1} + \omega_3 \\ (k_y)_{k-1} + \omega_4 \\ \theta_{FOG} + (\delta_{FOG})_{k-1} + \omega_5 \\ (\delta_{FOG})_{k-1} + \omega_6 \end{bmatrix} \tag{3.6}$$

where ω_n are the random variables which represent the process noise.

Our measure equation may be computed considering the GPS measures the absolute cartesian position of the vehicle and the TCM2 provides the absolute heading (yaw) angle:

$$\vec{z}_k = \begin{bmatrix} x_{GPS} \\ y_{GPS} \\ \theta_{TCM\,2} \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_1 \\ y_{k-1} + v_2 \\ \theta_{k-1} + v_3 \end{bmatrix} \tag{3.7}$$

Jacobian matrices

For the Extended Kalman Filter we need to calculate the following matrices:

$$A_{i,j} = \frac{\partial f_i}{\partial x_j}, \quad W_{i,j} = \frac{\partial f_i}{\partial \omega_j}, \quad H_{i,j} = \frac{\partial h_i}{\partial x_j}, \quad V_{i,j} = \frac{\partial h_i}{\partial v_j} \tag{3.8}$$

Thus, in our case, we have:

$$A = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_x \cos(\theta_{k-1}) & -v_y \sin(\theta_{k-1}) & -(k_x)_{k-1} v_x \sin(\theta_{k-1}) - (k_y)_{k-1} v_y \cos(\theta_{k-1}) & 0 \\ 0 & 0 & 1 & T & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v_x \sin(\theta_{k-1}) & v_y \cos(\theta_{k-1}) & (k_x)_{k-1} v_x \cos(\theta_{k-1}) - (k_y)_{k-1} v_y \sin(\theta_{k-1}) & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

$$W = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.10}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{3.11}$$

$$V = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

Initial conditions and covariance matrices

The first estimation of the state vector is based on the first measures from the GPS, DVL and the TCM2, and assuming that the GPS and DVL have the same scale:

$$\vec{x} = \begin{bmatrix} x_{GPS} \\ v_x \cos(\theta_{TCM\,2}) - v_y \sin(\theta_{TCM\,2}) \\ y_{GPS} \\ v_x \sin(\theta_{TCM\,2}) + v_y \cos(\theta_{TCM\,2}) \\ 1 \\ 1 \\ \theta_{TCM\,2} \\ \theta_{FOG} - \theta_{TCM\,2} \end{bmatrix} \tag{3.13}$$

In order to give a reasonable estimate of the covariance matrices, let's consider the following considerations:

1) The DGPS accuracy is about 1m, and it may vary according to the position of the satellites. Hence, the noise variable for x and y are somehow correlated (if the error on x increases, also the error on y increases).
2) The accuracy of the DVL measure of the velocity w.r.t. the bottom is about 0.4 cm/s
3) The TCM2 measure is somewhat unreliable, with an error of about 10 degrees.
4) The FOG is the most reliable source, with an Angle Random Walk (noise) of $4^\circ / hr / \sqrt{Hz}$.
5) The FOG stability is about $1^\circ / hr$

Based on the above considerations, we can initialize the error covariance matrix P as follows:

$$P = \begin{bmatrix} 1^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.2^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001^2 \end{bmatrix} \tag{3.14}$$

The covariance matrix of the process noise is:

$$Q = \begin{bmatrix} 0.01^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.001^2 \end{bmatrix} \tag{3.15}$$

Finally, the covariance matrix of the measurement noise is:

$$R = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0.2^2 \end{bmatrix} \tag{3.16}$$

High-Level Control (HLC)

Project Leader(s): Dr. Giacomo Marani
Personnel: Dr. Giacomo Marani
Past Project Leader(s): Dr. Tae Won Kim , Dr. Junku Yuh, Dr. Kazuo Sugihara & Dr. Song K. Choi
Past Personnel: Mr. Side Zhao, Ms. Jing Nie & Mr. Zhi Yao

Objectives

HLC's objective is to develop a supervisory control module that will minimize human involvement in the control of the underwater vehicle and its manipulation tasks.

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

In the gradual passage from human tele-operated manipulation to autonomous intervention, the most noticeable aspect is the increase of the level of information exchanged between the system and the human supervisor. In teleoperation with ROVs, the user sends and receives low level information in order to directly set the position of the manipulator with the aid of a visual feedback.

As the system becomes more autonomous, the user may provide only a few higher level decisional commands, such as "unplug the connector", interacting only with a higher level task-description layer. The management of lower level functions (i.e. driving the motors to achieve a particular task) is left to the onboard system. The level of autonomy is related to the level of information needed by the system in performing the particular intervention.

With the above considerations in mind, the HLC module initially involved the development of high-level task planning where a mission is always composed of two parts: the goal and the method of accomplishment. In other words, "what do I need to do" and "how do I do it." Following this strategy, a new high-level architecture of vehicle control, named the Intelligent Task-Oriented Control Architecture (ITOCA), was developed for SAUVIM.

In phase III-B there was a major upgrade of this configuration. The high level control layer of both the manipulation and the navigation systems have been standardized and upgraded to a powerful custom programming language.

A software emulated CPU, where the mission control resides, hosts this new dedicated programming language developed in order to address the above issues [Marani05]. This language, suitable for real-time embedded control systems, offers at the same time flexibility, good performance, and simplicity in describing a generic complex task. Its layer abstraction approach allows an easy adaptation to the hardware-specific requirements of different platforms. For example, the same module can be found in the manipulator platform for describing a generic manipulation task and in the main navigation controller for driving the vehicle to the target area. The client-server approach allows the necessary communications between the arm and the navigation module.

The language is completely math-oriented and capable of symbolic manipulation of mathematical expressions. The last is an important distinctiveness from most of the currently available robot programming languages. The procedural approach has been chosen in order to enhance the performance while maintaining the flexibility required for executing complex tasks. It is particularly suitable for real-time embedded systems, where the interaction of a generic algorithm with the time is critical.

Introduction

In the world of computer-controlled autonomous systems the choice of an appropriate programming language must address a wide range of issues.
Many autonomous systems act in an unstructured and dynamic environment. Here, the language must have the necessary flexibility to react to that world using sensor information and the available actuators. The unpredictability of events requires that a generic control algorithm be interrupted at any time, in order to face inconsistent or incomprehensible inputs and preserve the safety of the system.
A generic language must have the capability of specifying the task via an abstraction level. For example a vehicle going to rescue a target should accept inputs like "place the target in the varm workspace" and so on. Because each of the above operations require a complex low-level behaviors, a good programming language should span different layers: a high-level layer, where the task description takes place; a medium-level layer for describing the controls algorithms and finally a low-level layer which interacts with the robot hardware.
The last issue, the interaction between the hardware and the programming language, is one of the major problems in attempting to universally unify a generic robot programming system. Many companies developed their own language suitable for a particular system, hence the fact that no uniform consensus has been given to a particular language, is not surprising.
Another very important issue that a robot programming language must address is the time interaction. A generic control system is usually hosted by a real-time operating system, with at least a periodic task running at a fixed sample time in order to correctly quantify the discrete-time blocks (e.g. integrators, derivators, etc.). The mid-layer of the language, where part of the control algorithm resides, must have the capability of synchronizing with the above sample time, monitoring the execution length to avoid exceeding the time-line.
Finally, for a large class of autonomous systems like underwater robots, it is necessary to organize the language subsystem in a client-server architecture, in order to separate the human interface with the execution layer. In fact, they may have to reside in separate environments (e.g. the vehicle and the ground station).
The problem of structuring control functions in a multilayer structure has been dealt with in several works (Albus, 1987, Putz, 1992). In the NASREM architecture developed by Albus, a theoretical model was proposed consisting of six basic elements: actuators, sensors, sensory processing, world modeling, behavior generation, and value judgment. These elements are integrated into a hierarchical system architecture.
In SAUVIM we introduce two new high-level programming language (**SPL**, SAUVIM Programming Language, and **APL**, Arm Programming Language), developed at the Autonomous Systems Laboratory of the University of Hawaii.

Robot programming environments

Text-based control-specific languages are still the most common method of controlling industrial robots. An extensive review and classification has been done, as, for example, in MacDonald (2003), Biggs (2003) and Pembeci (2002).
Most of these programming languages have been very simple, with a BASIC-like syntax and simple commands for controlling robot behavior. Their biggest problem is the lack of a universal standard from robot manufacturer. Often robot manufacturers also provide a simulation environment.
The recent works in text-based systems have diverged from these robot-specific languages to develop more general purpose high-level programming languages suitable for any robot. Typically, this involves extending existing languages such as C++ (Dai, 2002), Java (Hardin, 2002, Kanayama, 2000), and Haskell (Hudak, 1997). In our work, the main effort was to provide a complete math-oriented programming environment together with an abstraction layer for adapting the programming language to the hardware-specific requirements of different systems.

Procedural versus Object Oriented

The object-oriented methodology seems a trend of computer language and basis of complex software. While the procedural approach divides problem into tasks to be performed, the object oriented approach decomposes a problem in terms of objects, and their attributes. OO is preferable for handling complex problems where the code must be error free.
However a framework for OO programming introduces some overheads at the execution stage. In robotics, the embedded control system architecture must follow strict time requirements. This is particularly important, for example, at the layer where tasks are transformed into movement of effectors, where the real-time interaction of the system with the environment is the most important issue.
Moreover embedded systems often have limited resources, like processor speed and memory, and their consumption should be limited as much as possible.
For this reason our choice was the procedural approach, at least for the lowest layers, where our work is focused. Future development of SPL, plans to introduce an object-oriented architecture for non real-time high-level algorithms, besides the procedural structure. At this aim, as mentioned later, it is possible to partition the planning problem into a hierarchy of levels with different temporal planning horizon within the SPL code.

SPL: Overview

The Sauvim Programming Language subsystem has been developed using the well-known tools Lex (Lesk , 1986) and Yacc (Johnson, 1979), a lexical analyzer and parser generator. Like C, Fortran, Pascal, Basic, and so on, SPL is a procedural language, in the sense that it consists of a sequence of commands, which are executed strictly one after the other. Like the other procedural languages, the code may be organized into procedures and libraries, which simplifies the separation of the high level (task oriented) layer from the mid-level layer. As a matter of fact, the latter consists of a set of procedures for attaining particular behaviors.

SPL is not strongly typed like C and Pascal and no declarations are required. It is more like Basic and Lisp in this respect. However types exist: type checking is done at run time and must be programmed explicitly.
It is interactive, and the programming language, although pseudo-compiled, is in turn interpreted by a software-emulated CPU. For this reason it is not suitable for running numerically intensive programs because of the virtual CPU overhead. Therefore the most computationally expensive algorithms have been implemented in a different layer (the MLC in fig. 3) and are accessible via the SPL abstraction layer, as better explained ahead.
Another important feature of the Arm Programming Language is the parallelization of servers. This option allows the spanning of different "processes" running simultaneously with the possibility of mutual interruption in case of particular events. The parallelization is very important when running, for example, health-monitoring procedures or for handling any kind of exceptions.
Finally the language is math-oriented and offers the possibility of symbolic manipulation of expressions, arrays and/or matrices.

Programming in SPL

Statements: Assignment, Conditional, Loop

The SPL syntax for the assignment is taken from Algol 60. The assignment statement looks like:

```
name := expr;
```

where `expr` is any expression and `name` is a variable name. The main difference between SPL and traditional programming languages is that the generic identifier is generally an entity that, if not assigned, stands for itself. In other words it is a *symbol*. Symbols are used to represent unknowns in equations, variables, indices, etc. Consider the assignment statement:

```
P := x^2 + 4*x + 4;
```

Here the identifier `P` has been assigned the formula `x^2 + 4x + 4`. The identifier `x` has not been assigned a value: it is just a symbol, an unknown. This assignment automatically sets the type of `P` to *expression*. The identifier `P` is now like a programming variable, and its value can be used in subsequent calculations just like a normal programming variable.
The conditional statement has the following syntax:

```
if expr then statseq
[ else statseq ]
end if
```

where *`statseq`* is a sequence of statements separated by semi-colons and `[...]` denotes an optional part. A typical `if` statement would be:

```
if x < 0 then -1; else 1; end if
```

The `for` statement has the following syntax:

```
for name from expr to expr do
   statseq
end do
```

Finally a conditional loop can be constructed according to the following syntax:
Table 1. SPL Data Types

null	numeric	name	matrix
intvector	+	*	^
builtinfcn	function	exprseq	stmtseq
executable	coeffterm	unknown	repeat
integer	list	indexed	if
ifelse	bool	asm	lrstack1
while	for	procedure	nameseq
string	uneval	eval	range
<	and	=	not
or			

```
while expr do
   statseq
end do
```

The loop is executed while the condition *expr* evaluates to `true`.

Data Types

Although it is not necessary to declare the type, a generic variable belongs to the class type corresponding its content. Many pre-defined types exist in SPL: a type can be any one listed in table 1. Type-checking can be realized at run-time. For instance, the expressions

```
type(x+y,"+");
type(1.2345,"string");
```

return respectively `true` and `false`. Types can be structured together in lists, similarly to the *struct* of C language. The resulting list object is a single entity composed of sub-objects of different types. This is necessary when working, for example, with map, search and map theory.
The most relevant is probably the `matrix` type. A matrix is a generic 2-D array, which in SPL can be constructed with the following syntax:

```
M := matrix(nRows, nCols,[[R1],[R2],...]);
```

where `nRows` and `nCols` are the dimension of the array and *Ri*, the i-th row, is a sequence of entries separated by commas. For example, a generic column vector can be defined as:

```
q := matrix(3,1,[[q1],[q2],[q3]]);
```

Procedures

One of the principal SPL tools used to customize and extend SPL's capabilities is the procedure.
In general, an SPL procedure has the following syntax:

```
proc (NameSequence)
   [local nameseq;]
   [global nameseq;]
   statseq
end
```

where *nameseq* is a sequence of symbols separated by commas, and *statseq* is a sequence of statements separated by colons or semicolons. The scope of visibility of local variables is within the procedure, while global variable are visible within the main workspace and within all the procedures in all the instantiated parallel servers. This offers a way to exchange data in between the spawned processes.
A procedure definition is a valid expression that can be assigned to a name. As an example, the command below assigns to the symbol `f` a user-defined procedure that adds its two arguments, `x` and `y`:

```
f := proc(x, y)
   x + y;
end proc;
```

This procedure has two parameters `x` and `y`. It has no local variables and only one statement. The value returned by the procedure is `x+y`. In general the value returned by a procedure is the last value computed. The following procedure call evaluates `f` with the arguments `3` and `5`:

```
f(3, 5);
```

It is also possible to invoke `f` with symbolic input:

```
f(Joint1, Joint2);
```

The hardware abstraction layer

The interaction between the hardware and the programming language is performed by a subset of built-in SPL procedures, completely user-definable in order to meet the requirements of the hardware device drivers.
Basically a built-in procedure for accessing the hardware level is an interface between SPL and any custom C/C++ code. It must be defined within the SPL source code before compiling the programming language system. Currently there are more than 80 built-in procedures and most of them perform some specific action like enabling the motors, getting the sensor information data, sending the motor velocity reference, etc. For example, in order to assign the variable `q` with the values of the joint angles, we can use the procedure `GetJointAngles`:

```
q := GetJointAngles();
```

which returns a 7-by-1 matrix containing the seven joint angles of the manipulator (in radians). Internally, the above procedure executes the following steps:

1) Creates an empty 7-by-1 matrix
2) Reads the IP-quadrature board via the device driver
3) Fills-up the 7 elements of the matrix with the above values
4) Returns the matrix

The internal C++ interface of SPL simplifies the writing process of built-in function with a large amount of classes, member functions and operators.

Memory management

Internally, a generic object is represented by a syntax tree of nodes (see fig. 5). Each node has an assigned type, a data field and an operand list. Data are interpreted according to the type of the node, and internally are stored in a special array with dynamic length and type. For example, the data field of a node of type `numeric` contains the numeric value of the node (in double-precision), while the operand field is empty.
Conversely, a node of type `function` contains no data and a variable number of operands (the function arguments). A function can be for example the addition operator: in this case the operands are the addends. Each addend can be, in turn, any generic node. This allows the easy handling of expressions in a symbolic form. The C++ interface provides all the necessary support for creating the generic node and manipulating its type, data and operand fields.
The creation/destruction process is assisted by a management algorithm which allows optimizing the allocated memory and avoiding memory leaks. Inside SPL, each node is stored in a dynamic array called *memory* in fig. 4. Each node is referenced by a special class that keeps track of the number of references to the node itself. When the node has no more references, it is marked as free and can be used for new contests. This allows the efficient reuse of the already allocated nodes, keeping the size of the dynamic memory to the indispensable minimum. This is very important when the server is running in embedded systems with limited amount of resources.
Fig. 4 also shows the basic concept that SPL uses for the execution phase. A software-emulated virtual CPU is responsible to fetch and execute the SPL statements according to the following steps:

1) The executable, consisting of an op-code table and a tree of data nodes, is loaded into the state machine system of fig. 4. More precisely, the op-code table is stored in the *executable table* while the set of data nodes is loaded into the *memory*.
2) The virtual CPU fetches the next available op-code from the current op-code table.
3) The required arguments involved in the execution of the current statement are pushed from the memory to the stack.
4) The CPU executes the current op-code, popping the arguments and pushing the result on the stack.

The above concept is replicated in parallel for each instantiated server. At the end of execution, after the last op-code, the CPU enters an *idle* status.

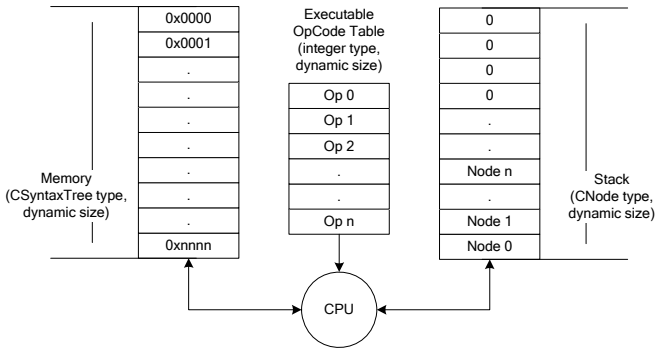


Fig. 4. Internal architecture: execution of statements via the software emulated CPU.

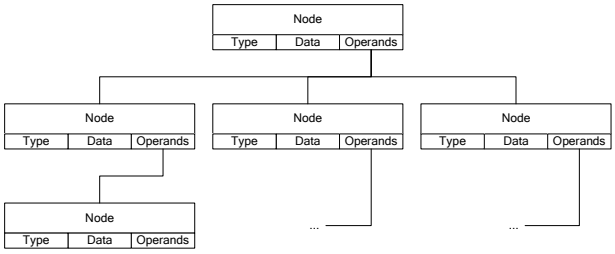


Fig. 5. Internal representation of objects: the syntax tree.

Client-server architecture

As earlier noted, the architecture of SPL is client-server oriented: the human interface (workspace input) and the execution layer reside in separate environments, communicating via a special protocol over TCP-IP (fig. 6). This allows the accomplishment of one of the main requirement of the NASREM model: with workspace input, the human operator can take over any layer at any time.
The client is generally a command-line input interface which allows loading libraries, executing statements and/or controlling the execution of the server. Fig. 7 shows a snapshot of our console implementation. The operations involved on the client-side are the followings:

1) Preprocess the source code.
2) Compile and create the executable.
3) Encode the executable.
4) Send the encoded executable via TCP-IP to the SPL server.

On the server-side, once the encoded executable has been received, the following operations take place:

1) Decode the executable.
2) Load the executable in the virtual CPU.
3) Execute.

One important feature of the overall programming language system is the possibility of using more than a single client. This allows for example, monitoring from a secondary client the overall behavior of the primary client, which may reside inside a separate autonomous system. For example, in our autonomous underwater vehicle, the primary client is inside the vehicle CPU, which sends the arm the operation commands. On the ground side, the user can monitor, with a secondary client, the behavior of the arm and eventually take control over the main CPU.
The next subsection summarizes the protocol used for exchanging data over TCP-IP.

The xBus Communication Layer

xBus is a TCP-IP based client-server communication system. The server can accept any number of client connections and each one can count on an error-robust communication protocol capable of auto-reconnection in case of a temporary network failure. This is important in a hostile environment, where the communication media does not allow safe and durable connections (acoustic modems).
Most of the network-based servers (such as FTP servers or HTTP servers) use a multi-thread approach for handling each client connection. xBus, on the other side, uses a different concept in order to accomplish the requirements of the SPL server. As a matter of fact, the last is shared between each connection and a parallel multi-thread approach would result in synchronization problems.
Internally xBus Server is a meta-state machine, or a set of finite state machines. These machines, one for each client connection, are called sequentially so that each one can request a different command execution to the parser. It is matter of the parser allowing or not, the execution of each command, according to its priority with respect to the already running ones.
A presettable internal timer generates an error if any reading or writing operation can not be executed within a certain amount of time. The error results in a forced disconnection followed by a reconnection attempt by the client. Upon any disconnection (graceful or forced), the correspondent server state machine is destroyed and removed from the machine list. Forced disconnection may happen even for other kind of socket errors (for example when losing the carrier of the acoustic modem during a mission), and are always followed by a reconnection attempt. For example, during the execution of any task, it is possible to unplug and successively re-plug the network cable: the only consequence is the loss of control by the client during the time that the cable is disconnected.
The kernels of both xBus server and client are written in Ansi-C language, including its vectorial state machine. This allows easily compiling the source code on a different platform, such as Windows (for the client or a generic simulation server) or VxWorks (the actual arm controller).

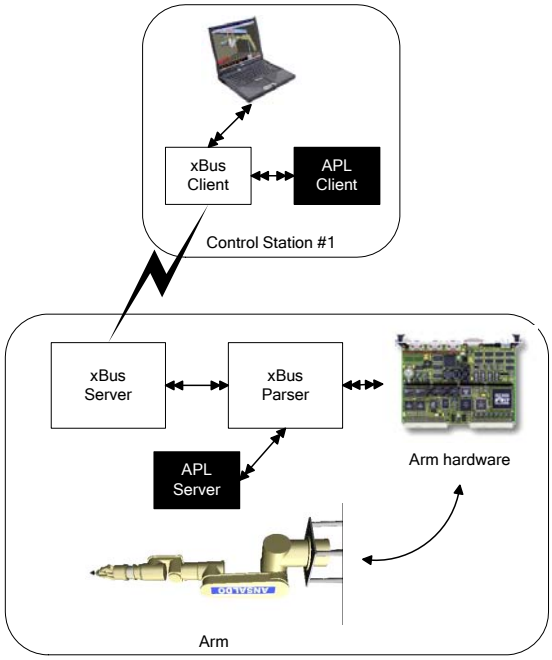


Fig. 6. The client-server architecture.

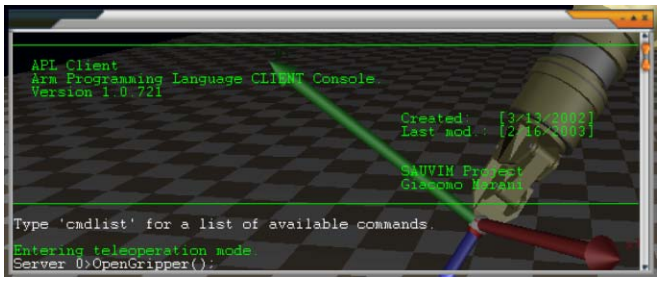


Fig. 7. The SPL client console.

Application example

In order to test and validate our programming language system we wrote a set of procedures for undocking the vehicle from the pier. After writing an opportune set of procedures, the task-level description procedure is:

```
// Undock ()
//_____________________________________________________________________
// Description.
// Undocks SAUVIM from the launch point.
Undock := proc()
      global DockAngle;
      local p1;

      print("$$$Warning: initiating Sauvim undocking sequence.\n");
      SetDefaultController();
      SetControllerAxis_xyy();
      p1 := matrix(6,1,[[0],[0],[DockAngle],[1],[-4.0],[-0.5]]);
      MoveTo(p1);
      p1 := matrix(6,1,[[0],[0],[0],[10],[-4.0],[-0.5]]);
      MoveTo(p1);
      print("$$$Undocking complete..\n");
end proc:
```

Future Tasks (Phase III-B Tasks)

- Introduction of a third high-level unit with a third programming language and higher priority over SPL and APL
- Implementation of more high level commands, especially network-based.

Virtual Environment (VE)

Project Leader(s): Dr. Giacomo Marani
Personnel: Dr. Giacomo Marani
Past Project Leader(s): Dr. Song K. Choi, Dr. Kazuo Sugihara, Dr. Stephen Itoga & Mr. Scott Menor
Past Personnel: Mr. Alexander Nip, Mr. Zhenyu Yang, Mr. Jiwen Liu, Mr. Steve Timcho, Ms. Lori Yokota, Ms. Jennifer Saito, Mr. Brandon Higa, Mr. Xiandong Su, Mr. Alberto Brunete, Ms. Tammy Yamauchi & Mr. Jeffery P. Yee

Objectives

The VE is aimed at developing a supervisory monitoring system for SAUVIM to smoothly and realistically integrate mapping data with on-line sensory information even in the case of low bandwidth. It is the evolution of the old idea of the Predictive Virtual Environment, described in the previous reports of SAUVIM, into a more advanced system collecting also the virtual manipulator and the SAUVIM control interface through direct interaction with the virtual environment.

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

The PVE was originally aimed at developing a supervisory monitoring system for SAUVIM to smoothly and realistically integrate mapping data with on-line sensory information even in the midst of delayed and limited information. The development for the PVE has been modular. The various modules were: the SAUVIM Simulation Software (SSS); the SAUVIM Video Overlay Software (SVOS); the Communication Software (CS); and the artificial neural network (ANN) Video Prediction Software (VPS). In the Phases I and II of SAUVIM the SSS has been upgraded from its *Version 1* to *Version 1.1*, which includes the incorporation of a Magellan spaceball mouse, an accurate 3D graphical model of SAUVIM and the Maris 7080 manipulator, scene-smoothing methods using interpolation techniques, and an easy-to-use user interface. The SVOS was developed to overlay video images of the seafloor (texture and color) to the graphic images to provide a more accurate monitoring of the vehicle, manipulator and environment. The CS for SAUVIM was an extension of the NSF's DVECS (Distributed Virtual Environment Collaborative Simulator) project. At that time, the DVECS system used a cellular phone to communicate the vehicle data from the test-site to the monitoring computer located on campus for data fusion. Experiments have been conducted with the ODIN AUV. The experiments of ODIN were projected via an ElectroHome Marquee 8500 CRT projector coupled with multiple Stereographics (SG) emitters and SG CrystalEyes glasses. Finally, the VPS has been tested, and, although in its early stage, with positive results.

Successively, due to the high maintenance costs of SGI workstations, the overall virtual reality and monitoring system, which includes the video prediction, has been transformed to a much more stable and inexpensive personal computing system, taking advantage of the

emerging market of high performance hardware video accelerators (mostly targeted to PC games).

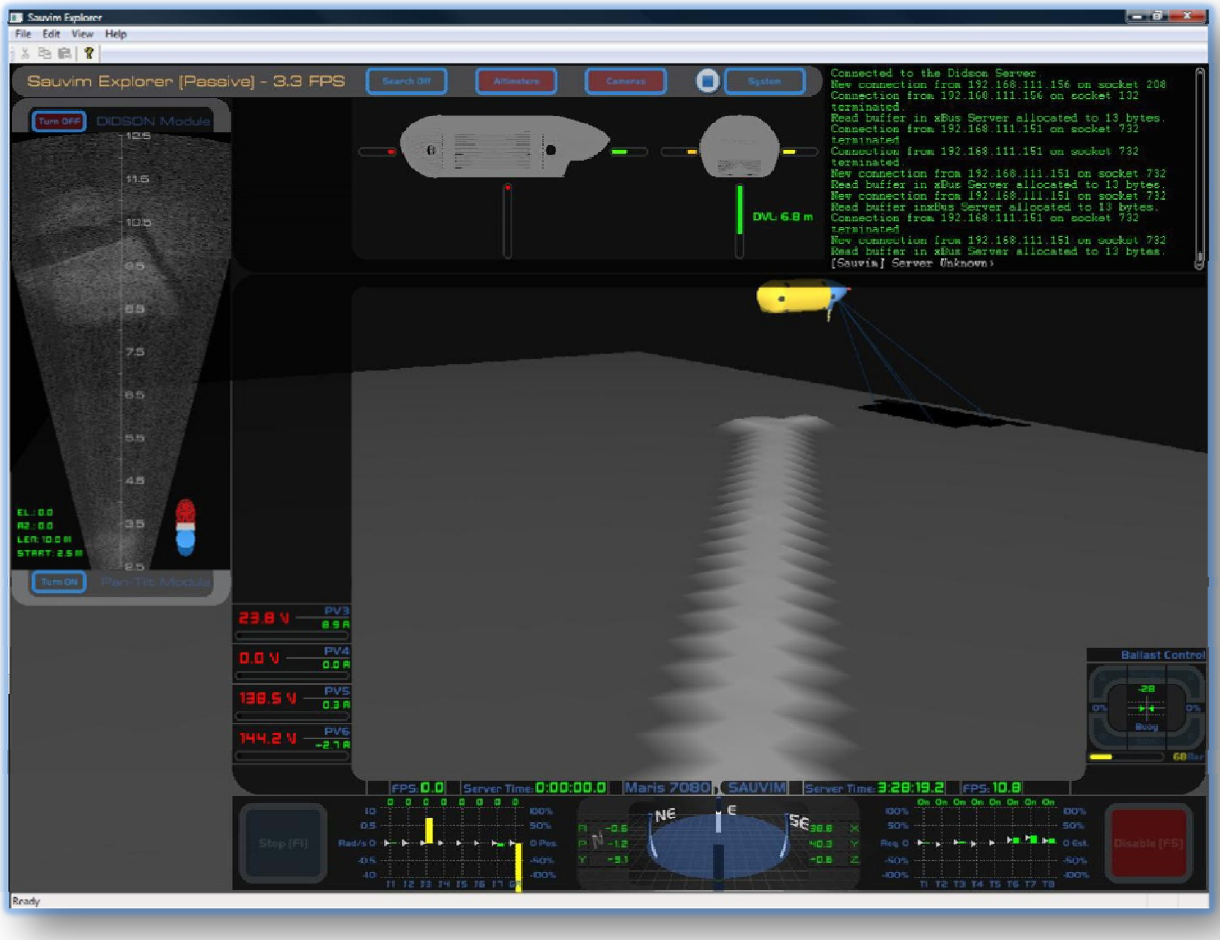


Figure VE-1: Sauvim Explorer

MarisGL was, during the Phase II, the preliminary version of the virtual environment targeted to the MARIS 7080 Manipulator and making use of a standard OpenGL PC video accelerator. During the Phase III-A the application was extended in order to introduce the vehicle model, mainly for collision avoidance verification. But the most important transition toward the global virtual environment happened in the current Phase III-B.

Here, the name of the application, once targeted to visualize only the configuration of the arm, has been changed to Sauvim Explorer (Figure VE-1). Sauvim Explorer collects in a unified application the data from all the sensors of SAUVIM, including data from the DIDSON that can be overlaid over the graphical reconstruction of the floor.

It also hosts the remote console clients for both the Arm Programming Language and the Sauvim Programming Language servers, and may act as remote control (ROV mode) when

a sufficient bandwidth channel is present. At this aim Sauvim Explorer contains software interface with several input device hardware, including 6 DOF space controllers.

This represents an enormous step forward toward the unification of the whole system, since it required a huge effort on the standardization of the communication protocol between every module of SAUVIM (sensors, actuators, controllers…). With this modular approach it is now extremely easy to add further sensor modules to SAUVIM and add their input and outputs to the SE application with a minimal effort.

The following is the summary of the major key points:

- Unified interface for SAUVIM and MARIS Manipulator
- Support for SPL (Sauvim Programming Language) and APL (Arm Programming Language) clients in the same console
- Integration of the DIDSON interface
- Integration of the altimeters
- Integration of the pan-tilt control

Following some screenshots of the actual Virtual Reality interface.

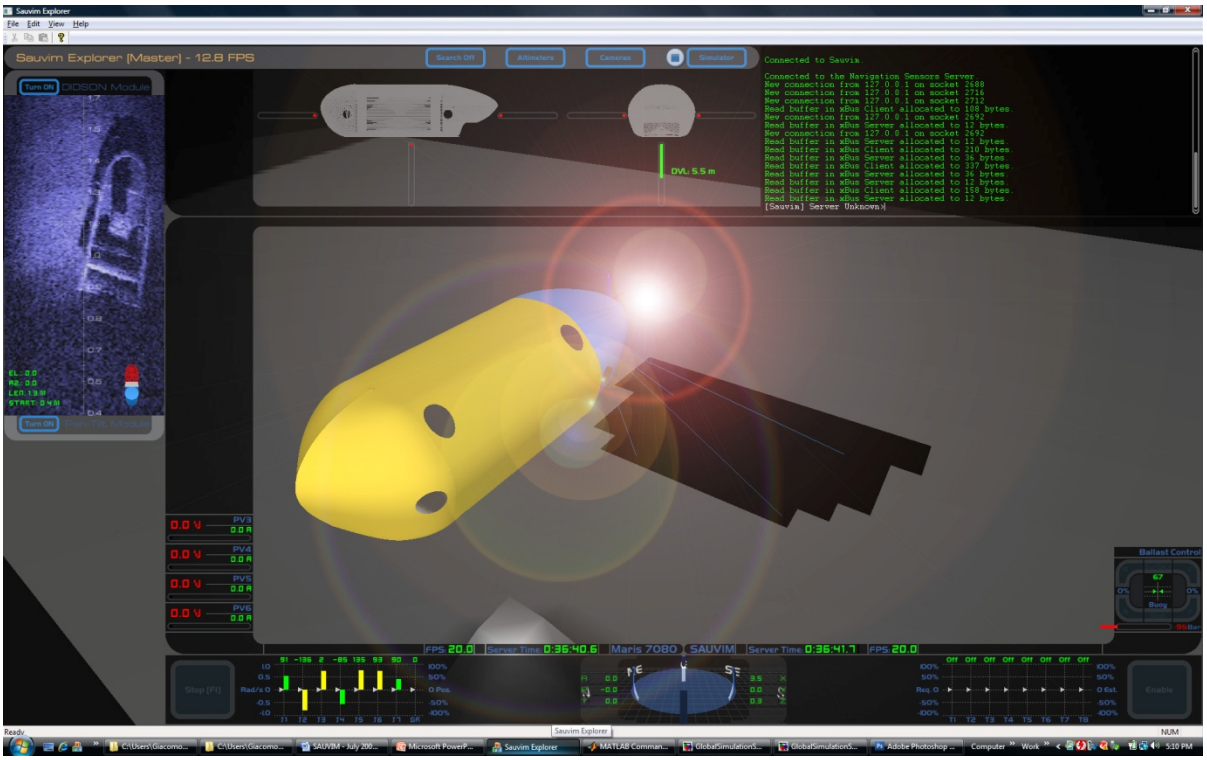


Figure VE-2: General Interface

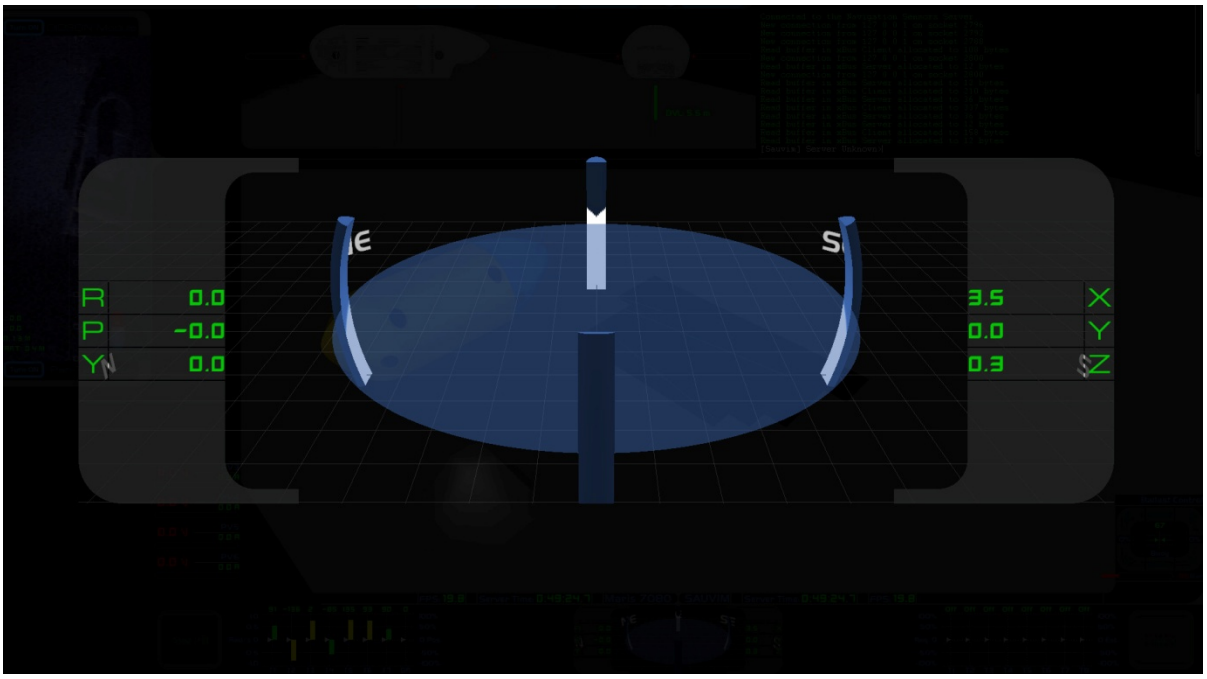


Figure VE-3: The Generalized Position display

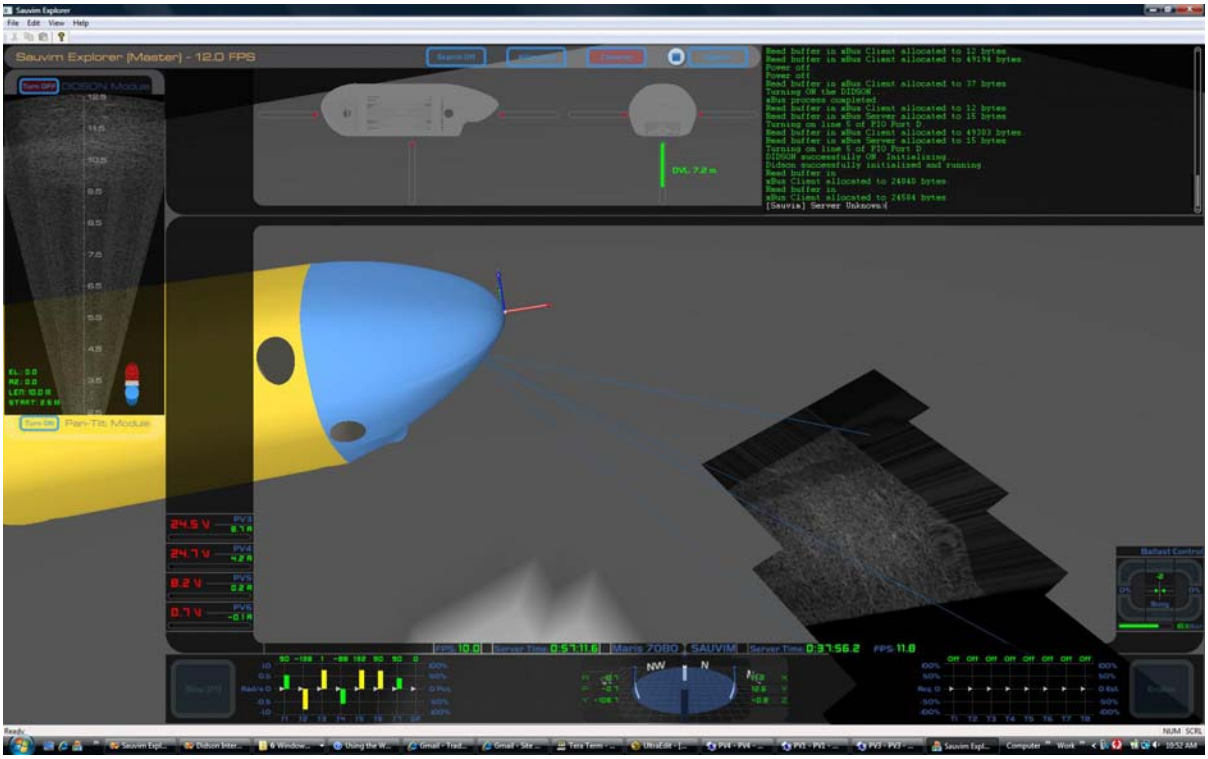


Figure VE-4: Real-time terrain generator (height mapping) for the virtual reconstruction of the ocean floor, with real-time overlay of the DIDSON image

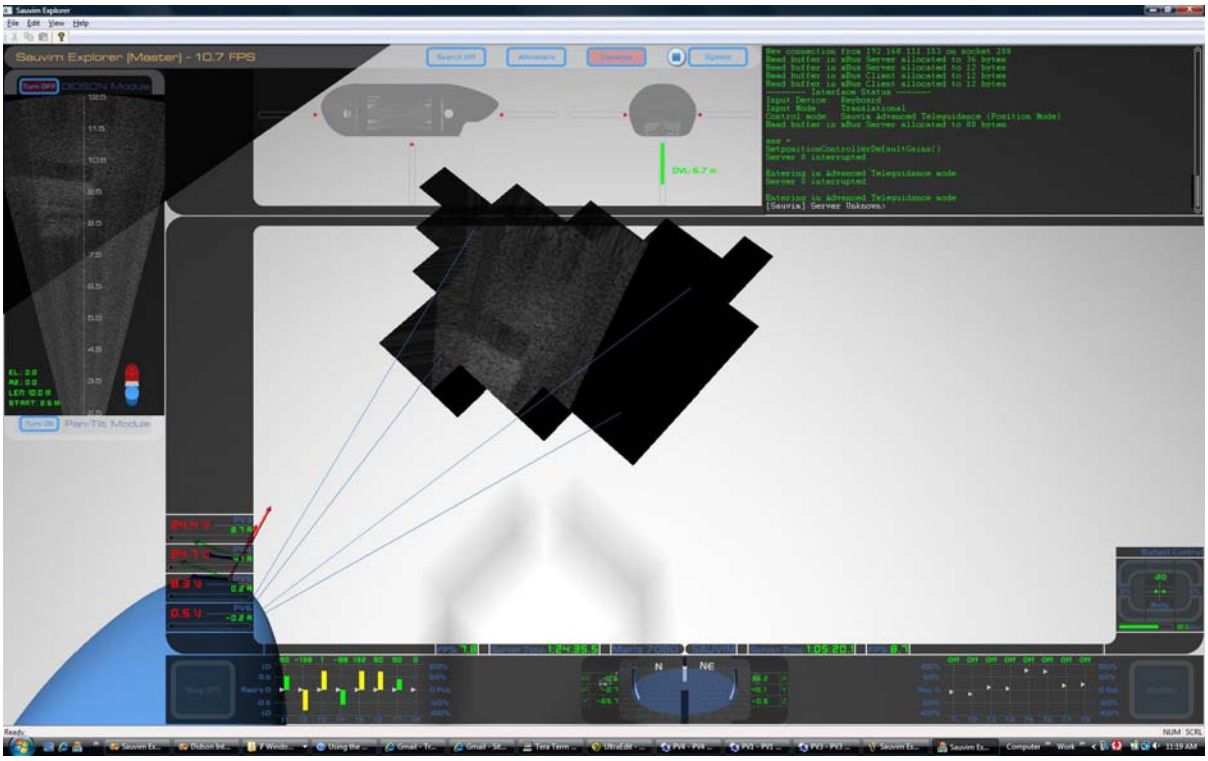


Figure VE-5: Support for 6 DOF motion controller devices, for an alternate driving solution for both the vehicle and manipulator (in case of teleoperation/teleguidance)

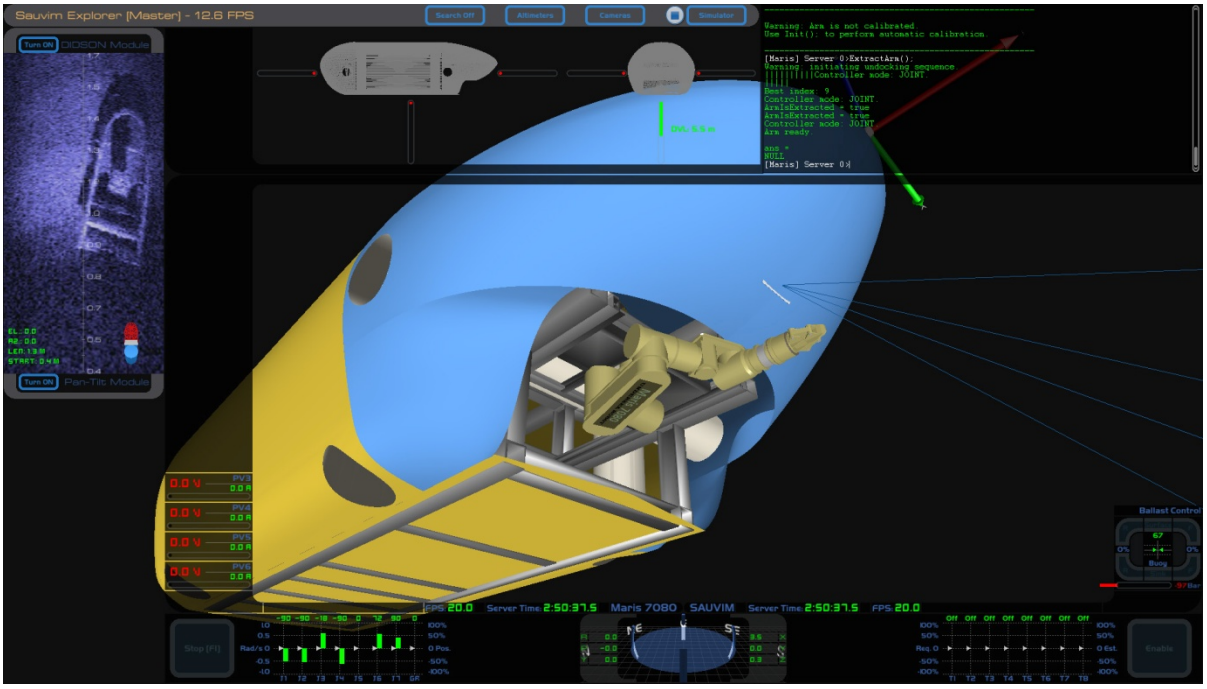


Figure VE-6: Real-time link with the Arm subsystem

SAUVIM Design (SD)

Project Leader(s): Dr. Giacomo Marani, Dr. Song K. Choi
Past Project Leader(s): Dr. Curtis S. Ikehara, Dr. Junku Yuh, Dr. Mehrdad Ghasemi Nejhad, Dr. Gary McMurtry, Dr. Pan-Mook Lee, Dr. Farzad Masheyekhi, Dr. Gyoung H. Kim, Mr. Gus Coutsourakis, Mr. Oliver T. Easterday & Mr. Michael E. West

The main technical development of the SD group is described in the following sections: Reliable, Distributed Control, Mission Sensor Package, Hydrodynamic Drag Coefficient Analysis, Mechanical Analysis & Fabrication and Mechanical-Electrical Design. Many of the developments relative to the SD group have been competed in the previous phases. However the Phase III-B has seen substantial changes in the Reliable, Distributed Control, here described.

Reliable Distributed Control (RDC)

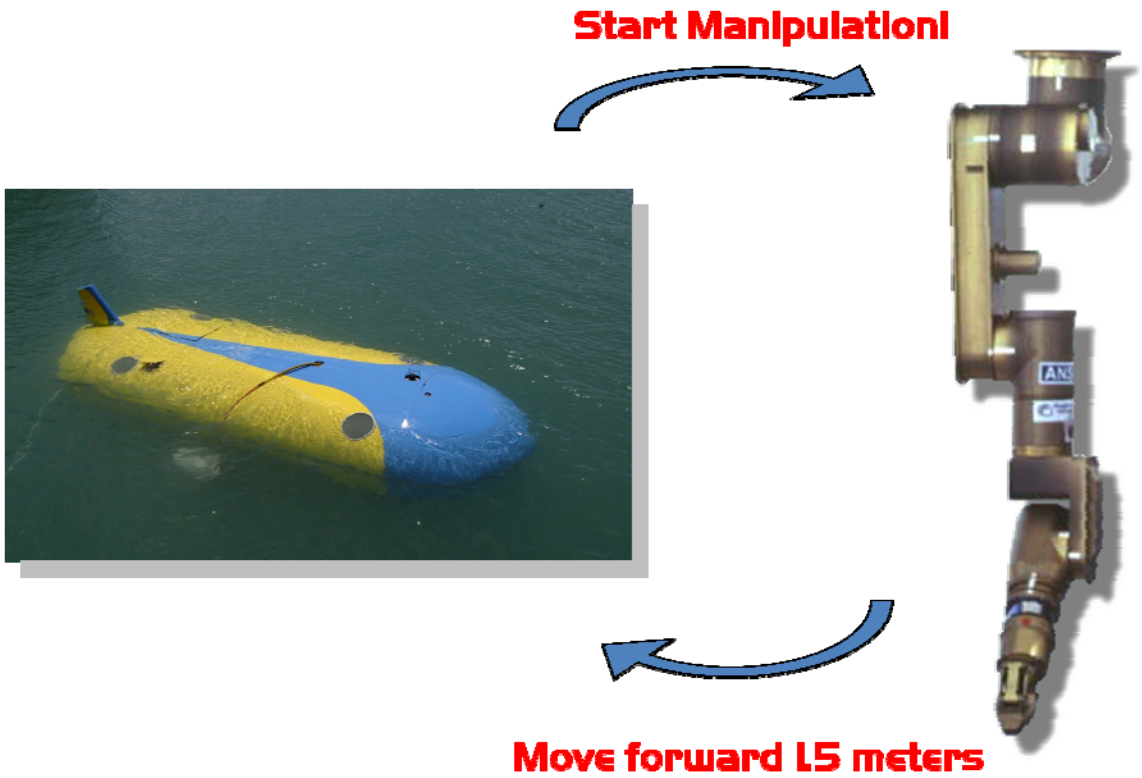
Project Leader(s): Dr. Giacomo Marani
Personnel: Dr. Giacomo Marani
Past Project Leader(s): Dr. Tae Won Kim, Dr. Pan-Mook Lee, Dr. Curtis S. Ikehara, Dr. Song K. Choi & Dr. Gyoung H. Kim
Past Personnel: Mr. Jang-Won Lee, Mr. Michael West, Mr. Tuan M. Hyunh, Dr. Hyun Taek Choi, Mr. Alberto Brunete & Mr. Alexander Nip

Objectives

The objective is to develop a reliable & efficient computing architecture for signal and algorithmic processes of the entire SAUVIM system.

Current Status (Tasks Completed during 12/15/2005 - 12/20/2008)

The SAUVIM Phase III-B has seen a completer reorganization of the Main SAUVIM control system. The goal was to unify the communication language between different modules, for increasing the level of autonomy of SAUVIM.



This is extremely important for sharing information between different subsystems, like the manipulator and the navigation units in the above figure.

The new real-time architecture of SAUVIM

Figure Figure RDC-1 showthe new SAUVIM control organization.

The new architecture plan for the SAUVIM platform has been developed with a heavy emphasis to autonomy and global information sharing. It has several similarities to the backseat driver paradigm [Ben07], which has been implemented on a number of platforms (e.g., Bluefin, Hydroid, and Ocean Server). The paradigm refers to a division between "low-level" control and "high-level" control on the vehicle, with most likely the former residing on the vehicle's main computer and the latter residing on a computer in a payload section that can be physically swapped out of the vehicle. The low-level control is also referred to as "vehicle control" and the high-level control as "mission control". Here, the architecture that coordinates the set of software modules collectively comprising the `backseat-driver` system running in the payload has been implemented using MOOS[1]

SAUVIM uses a similar configuration, with a precise role separation between high-level or mission control (in the `backseat`) and low-level or vehicle control (in the `front-seat`). This separation has been implemented with a dedicated software environment for autonomous systems. The mission control system [backseat] is basically a software-emulated CPU that runs a custom programming language specially created in order to simplify high-level operation and algebraic manipulations at the same time.
Since it is a software-emulated CPU, it can be compiled within the main vehicle computer while still maintaining the virtual separation between the mission control and the vehicle control [front-seat]. The hardware resides within an abstraction layer, and the entire language can be easily re-adapted to a different hardware layer, given a precise and standard specification for the interface procedures. Figure 3 shows this concept implemented for the vehicle navigation system.

Within the mission control layer, another very important issue that the programming language for autonomous systems must address is the time interaction. A generic control system is usually hosted by a real-time operating system, with at least a periodic task running at a fixed sample time in order to correctly quantify the discrete-time blocks (e.g. integrators, derivators, etc.). The mid-layer of the language, where part of control algorithm may reside, must have the capability of synchronizing with the above sample time while monitoring the execution length for avoiding on exceeding the time-line. This is easily achieved since in our approach the local backseat resides within the same vehicle control (Main Vehicle Computer, MVC) and the software-emulated CPU can be looped directly within the main control loop. This has the immediate advantage of performing additional high-level operation like real-time tracking of time dependant trajectories.

This distributed programming environment for autonomous systems is completely written in ANSI-compliant C and C++, and can be cross-complied for different platforms (VxWorks, Windows, Unix…). This make it possible to break the environment into separate parts, the software-emulated CPU and the code generator ("complier"): the execution CPU can run

[1] Mission Oriented Operating Suite, developed by Paul Newman at the MIT, Department of Ocean Engineering, http://www.robots.ox.ac.uk/~pnewman/TheMOOS/

inside the real-time controller (for instance running a VxWorks operating system) while the compiler may reside on a remote platform such as Windows or Unix, linked via the communication system. Figure 3 shows the case where the remote client is a personal computer residing externally (at least when the communication link with the vehicle is available).

This configuration is duplicated for the manipulator and linked together with the SAUVIM navigation system through the main communication layer xBus.

Distributed Control: the data exchange bus

SAUVIM uses a client-server approach for delivering information from and to each distributed module.
Each subsystem (as a backset module or a generic sensor) embeds a custom TCP-IP client-server communication system (`xBus`, see [Marani05]). Within this architecture, every server can deliver the requested information on-demand to any number of clients, and this configuration allows a different utilization of the bandwidth, since every data is broadcasted only on demand.
This approach is similar to the Publish-Subscribe Middleware paradigm [Ben07], where the term `middleware` refers to the architecture software that coordinates the set of software modules collectively comprising the backseat-driver system running in the payload. Publish-subscribe middleware implements a community of modules communicating through a shared database process that accepts information voluntarily published by any other connected process and distributes particular information to any such process that subscribes for updates to such information.
In the SAUVIM approach the information is not published by a central database, but every source acts as a server that may send only the requested information to the requesting client. The distributed client-server architecture also provides a security hand-shaking mechanism, which provides direct feedback on the execution of any instance of data exchange. This is particularly desirable in issuing security commands (such as for aborting the mission).

The SAUVIM Simulator

An immediate advantage of the client-server approach is that each module can be transparently substituted by a simulator, without affecting the structure of each backseat. This is done selecting, in each client side, the appropriate IP address of the server.
In our system, the vehicle model has been implemented via Simulink (The Mathworks, Inc). The communication server, the programming language server, the task space controller and the navigation controller have been compiled and embedded in a custom Simulink block.
Since the source code is essentially the same for the simulator and the actual system, this process allows testing and simulating every aspect of the control system before running it on the actual vehicle.

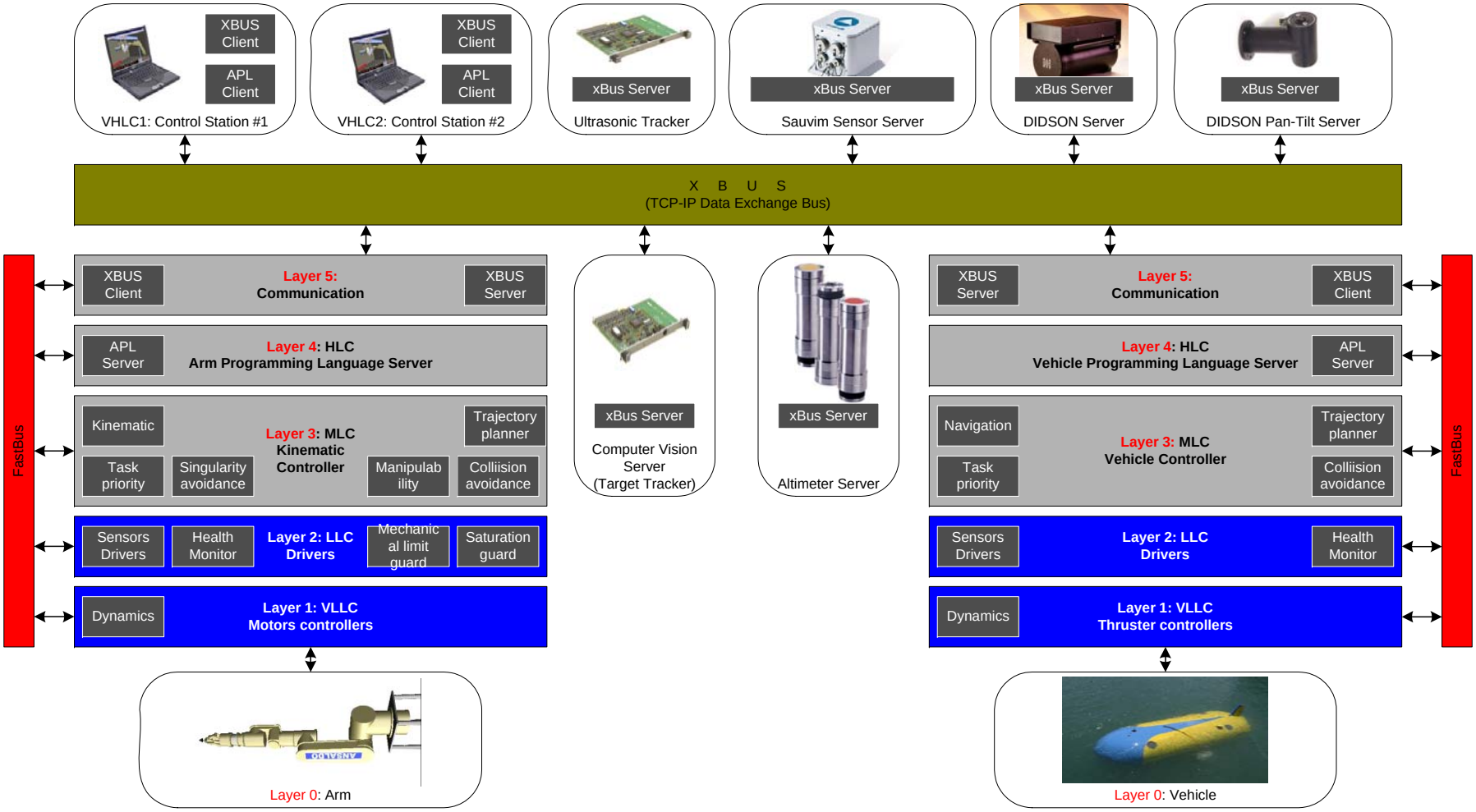


Figure RDC-1. The new SAUVIM control diagram.

Main SAUVIM sensors

Altimeter: Tritech PA200

SAUVIM will be equipped with seven range sonar sensors, Tritech PA200. One is for altitude (vertical) and the others are for range measuring. These sensors have RS-485 multi-drop serial communication interfaces. SAnd, star topology is used for physical connection, because it doesn't affect the rest of the connection, and it is easy to add and remove nodes. Table RDC-2 shows the specification of PA200 sensors.

Table RDC-1. Specification of Tritech PA200

Frequency and beam width	200 kHz and 20 degrees
Measurement range	100 meters
Operating depth	6800 meters
Input voltage	12 VDC
Interface	RS-485, 9600 bps, 8 data bits, 1 stop bit, no parity
Head RS-485 Termination	220 Ω (Sensor A only)
Command	*, or 'A', 'B', 'C', 'D', 'E', 'F', 'G'

Electronic Compass Sensor: TCM2

TCM2 is an electric compass sensor module. It has a three-axis magnetometer and two-axis tilt sensor. In addition to compass heading, the TCM2 supplies pitch, roll, magnetic field data and temperature information. This sensor can be used as a backup sensor for the AHRS-BA303 sensor. And, it also uses moving average and min/max cancellation methods to have noise immunity. The detailed specification of TCM2 is shown in Table RDC-3.

Table RDC-2. Specification of Precision Navigation TCM2

Heading information	Accuracy when level	±0.5° RMS
	Accuracy when tilted	±1° RMS
	Resolution	0.1°
	Repeatability	±0.1°
Tilt information	Accuracy	±0.2°
	Resolution	0.1°
	Repeatability	±0.2°
	Range	±20°
Magnetic field information	Accuracy	±0.2 μT
	Resolution	0.01 μT

	Repeatability	±0.2 μT
	Range	±80 μT
Temperature information (sensor is uncalibrated)	Accuracy after calibration	±1°C, ±2°F
	Resolution	1°C, 2°F
	Range	-20°C to 70°C
Power requirement	Supply voltage	+5 VDC regulated 6 to 18 VDC unregulated
	Current	Standard mode: 15-20 mA Low-power mode: 7-13 mA Sleep mode: 2.5 mA
Interface	Digital	RS-232C, NMEA0183
	Analog	0-5V linear, 19.53 mV resolution (256 discrete levels), 0-5 quadrature (sine and cosine)

Scan sonar: Imagenex 881 high resolution imaging sonar

The Imagenex sonar is an image scanning sonar. It will provide scanned images around the vehicle. The scanned images can be used for obstacle avoidance or target detecting. The sonar consists of two parts. One is a sonar module with a rotating sonar head. The other is a digital signal processing module, which processes sonar signal and transmits processed data via RS-485 interface. Two modules are connected with an oil-filled underwater cable. The processing module is connected to the pressure vessel of the navigation control system with a 4-conductor underwater cable. Table RDC-4 shows specification of the Imagenex 881 sonar. For the forward and backward scanning, two sonars will be installed at head and tail of vehicle. Since the communication speed for scan sonar is so fast that Pentium-based PC/104+ is used to handle communication data and image processing.

Table RDC-4. Specification of Imagenex 881

Frequency	675 kHz
Transducer	Imaging/profiling
Power supply	22 – 48 VDC at 1 Amp max.
Interface	RS-485 (115200 bps, 8 data bits, 1 stop bit, no parity)
Operating range	6000meters
Measurement range	5 – 200 meters 15 – 600 feet Default: 50m (150ft)
Sector size	Scan with angle Sector mode: 0 to 180° in 3° increments. Default: 180° Polar mode: 0 to 360° in 3° increments

	Default: 360°
Speed	Step size angle Slow: 0.3°/step Med: 0.6°/step Fast: 0.9°/step Faster: 1.2°/step Fastest: 2.4°/step Default: fast
Transmit pulse length	0 to 255 μs in 5 μs increments

DIDSON (Dual IDentification SONar)

DIDSON is a high-definition imaging sonar which gives near video quality images for inspection and identification of object underwater. It is a surrogate for optical systems in turbid water. The standard DIDSON operates at two frequencies (1.8MHz and 1.1MHz) and provides images of objects from 1 meter to over 30 meters in range. Details are described in Appendix RDC-E.

Table RDC-5. Specification of DIDSON

Mode	Detection Mode	Identification Mode
Operating frequency	1.1 MHz	1.8 MHz
Beam width	0.4° H × 14° V	0.3° H × 14° V
Number of beams	48	96
Max frame rate	4-21 frames/sec	
Field-of-view	29°	
Power consumption	30 W typical	
Weight	17.4 lb (Air), 2.2 lb (Water)	
Dimensions	31.0 cm × 20.6 cm × 17.1 cm	
Control	Ethernet	
Display up-link	Ethernet or NTSC video	
Maximum cable length	200ft (100/10 BaseT)	

Future Tasks (Phase III-B Tasks)

1) Migrate current S/W to new H/W environment
2) Upgrade system S/W
3) Upgrade TDL for complex task description
4) Program and run tasks with TDL in the field

Mission Package Sensors (MSP)

Project Leader(s): - none -
Personnel: - none -
Past Project Leader(s): Dr. Gary McMurtry, Dr. Song K. Choi & Mr. Oliver T. Easterday
Past Personnel: Mr. Yann Douyere, Mr. Alan Parsa & Mr. Max D. Cremer

Objectives

The SAUVIM Mission Sensor Package for Phase 1 is designed to provide semi-continuous records of AUV water depth (pressure), water temperature, conductivity, computed salinity, dissolved oxygen, pH and turbidity for at least eight hours. These parameters as well as the magnetic signature of the seafloor can be acquired by the SAUVIM in survey mode. In intervention mode, the Mission Sensor Package will provide AUV water depth (pressure) and the water temperature and compositional parameters at a selected seafloor target, including pumped samples from submarine seeps or vents.

Current Status (Tasks Completed)

The task has been completed in the previous phases. Refer to previous reports for its descriptions.

Hydrodynamic Drag Coefficient Analysis (HDCA)

Project Leader(s): - none -
Personnel: - none -
Past Project Leader(s): Dr. Song K. Choi, Dr. Farzad Masheyekhi, Dr. Junku Yuh, Dr. Curtis S. Ikehara & Mr. Oliver T. Easterday
Past Personnel: Mr. Brian S.C. Lau

Objectives

- Determination of the hydrodynamic coefficient via numerical solution of full Navier-Stokes equations using commercial CFD code, PHOENICS.
- Provide design recommendations for the vehicle fairing from the hydrodynamic results.
- Perform experiments to verify and confirm the CFD results.

Current Status (Tasks Completed)

The task has been completed in the previous phases. Refer to previous reports for its descriptions.

Mechanical Analysis and Fabrication (MAF)

Project Leader(s): Dr. Song K. Choi
Personnel: - none -
Past Project Leader(s): Dr. Mehrdad Ghasemi Nejhad & Mr. Oliver T. Easterday
Past Personnel: Dr. Ali Yousefpour, Mr. Eric Sung, Mr. Bruce Flegal, Mr. Robert Ng, Mr. Mark Uyema, Mr. Saeid Pourjalali, Ms. Melanie Yamauchi & Mr. Reid Takaiya

Objectives

Mechanical Analysis and Fabrication (MAF) group is responsible for designing, analyzing, manufacturing, and testing of pressure vessels and flooded fairing as well as analyzing the metallic frame of the vehicle.

Current Status (Tasks Completed)

The task has been completed in the previous phases. Refer to previous reports for its descriptions.

Mechanical-Electrical Design (MED)

Project Leader(s):	Dr. Song K. Choi, Dr. Giacomo Marani
Personnel:	Mr. Kaikala Rosa, Mr. Aaron Hanai, Mr. Christopher A. McLeod, Mr. Edgar Gongora, Mr. Scott Weatherwax, Mr. Patrick Simmons, Mr. Greg Tamasahi.
Past Project Leader(s):	Dr. Curtis S. Ikehara, Dr. Junku Yuh, Mr. Gus Coutsourakis, Mr. Oliver T. Easterday & Mr. Michael E. West
Past Personnel:	Mr. Ismael Medrano, Mr. Dante Julian, Mr. Stacy Hanson, Mr. Lawrence Wong, Mr. Mark Fujita, Mr. Dicson Aggabao, Mr. Szu-Min Chang, Ms. Colleen Kaku, Mr. Mike Hall, Mr. Tai Blechta, Mr. Scott Sufak, Mr. Keith Sunderlin, Mr. Clyde Campos, Mr. Richard Antunes, Mr. John Lee, Mr. Scott Sufak, Mr. Daniel Shnidman, Mr. Weston Fujii, Mr. John Lemmond & Ms. Elizabeth Shim

Objectives

Integrate mechanical and electrical components of the SAUVIM vehicle and provide vehicle infrastructure in terms of structure and power to support research aspects of SAUVIM AUV. One of the most relevant progress in the Phase III-B was the Thruster power system upgrade.

Current Status

Sauvim Sensor Server

Objective

Design and build a stand-alone sensor server system for SAUVIM.

Background & Project Description

Our design requirement was to build a stand-alone sensor system that could be integrated into SAUVIM. Previously, all sensor and navigation information was processed in the main navigation bottle. With the addition of the sensor server, all of the critical navigation data could be processed before being sent to the navigation computer, which frees up valuable CPU time. Another added feature to the sensor server is the introduction of the PHINS (Photonic Inertial Navigation System) sensor, which is a fiber optic gyro navigation system, produced by IXSEA. The PHINS is able to receive and integrate data from the GPS (global positioning system), DVL (Doppler velocity logger) and a pressure sensor to compute an accurate relative position.

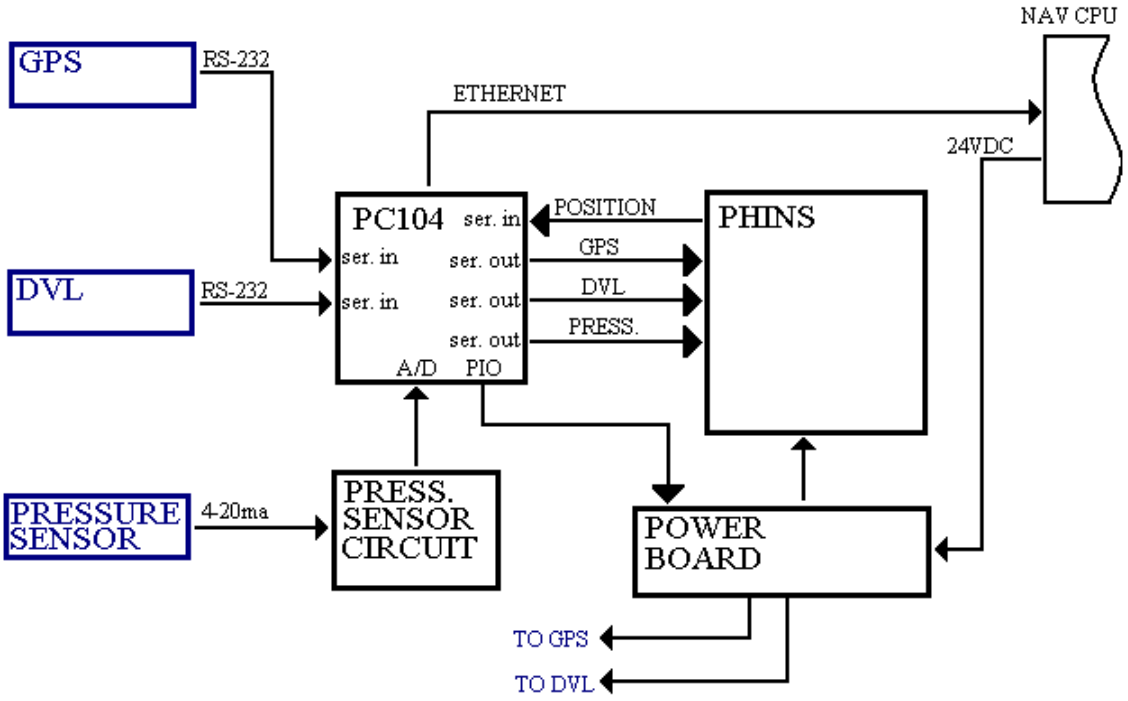


Figure 1: Block diagram of sensor server electronics

In addition to the PHINS, a PC104 mini computer was installed in the system to handle data between all of the sensors and the PHINS. By doing this, the data could be sampled and transmitted to the surface as a real-time update on the operator's user interface. An additional benefit to the PC104 is that it allows for a proper power up and shutdown of all sensors.

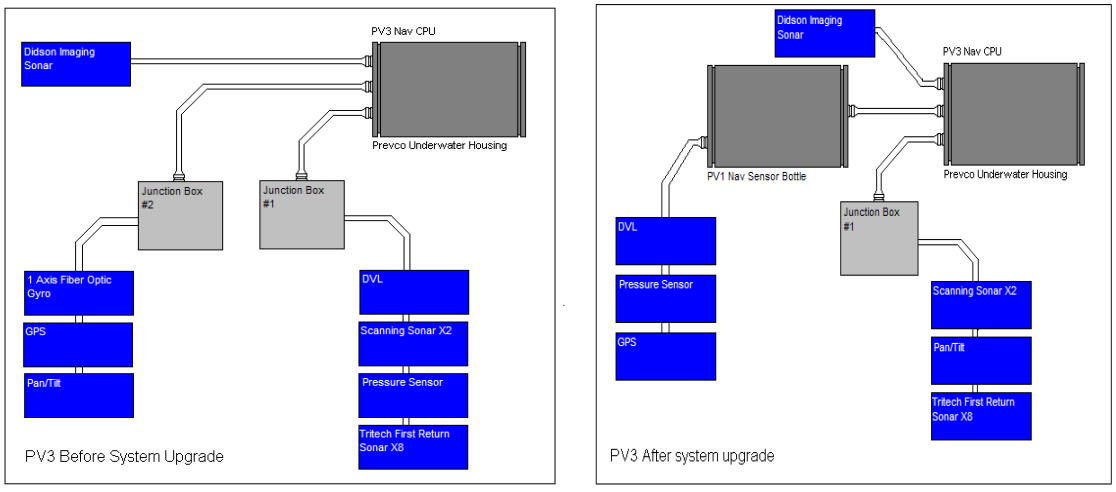


Figure 2: Physical layout of SAUVIM sensor system before and after modifications

The sensor server communicates to the navigation computer via 100mbps ethernet connection using the communication protocol called xBus*, developed by Dr. Giacomo Marani. With this protocol, any computer on the SAUVIM internal network can access the data at any time. All of the electronics are housed in a thirteen inch diameter aluminum housing constructed by Prevco SubSea housings. The internal electronics frame was constructed to fit in a nine inch housing to be able to transport the electronics into a full ocean depth titanium housing at a later date.

Figure 3: SAUVIM Sensor Server during construction

Thruster Power System Upgrade

Objective

Upgrade the existing battery system to increase mission time and available power to SAUVIM's thrusters.

Background & Project Description

The thrusters have a dedicated battery bank separate from the rest of systems on board SAUVIM. The battery bank provides power to SAUVIM's eight thrusters; 4 vertical, 2 lateral, and 2 horizontal. The 4 vertical thrusters are smaller than the lateral and horizontal and will take up to 7Amps each at full thrust. The horizontal and lateral thrusters can use up to30 Amps each at full thrust. All the thrusters are set to operate at 144~150VDC, therefore if all thrusters were run at full thrust simultaneously the maximum current could

potentially be as high as 148 Amps @ 144V which is approximately 21 kilowatts of power. The potential maximum current is not a likely scenario, but it is used as a worst case for the upgrade design.

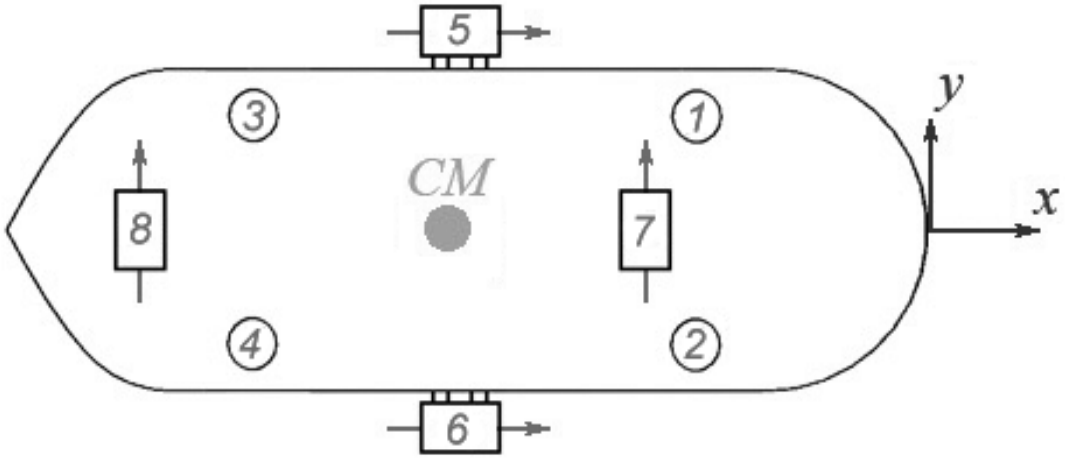


Figure 1 SAUVIM thruster configuration: 1-4 Vertical, 5-6 Horizontal, 7-8Lateral

The previous thruster battery system used series-connected 12V, 18Ah Pb-acid batteries. Six oil filled battery housings each held 4 batteries connected in series. This allowed the use of a standard 48V charger to charge the batteries. Three 48V batteries connected in series provided the 144V to one set of four thrusters and the other three 48V series batteries to the remaining four thrusters. However the limitations on operating time could be seen during tests, as the 18Ah batteries had a very short run time, approximately 3hrs. The 3 hours of run time came from the fact that the thrusters were current limited to 5 Amps for the vertical and 15 Amps for the horizontal and lateral. Furthermore, during tests the thrusters were not always running, and when they were they were not all running simultaneously. An upgrade to the battery system would increase mission times and would provide enough power to realize the full thrust capabilities.

Panasonic's rechargeable AHR-300H 3000mAhr, 1.2V, C-cell NiMH batteries were chosen for the battery upgrade because of their increased power density compared with Pb-Acid. Li-Ion batteries were also considered, but compared to NiMH the cost versus power made the NiMH batteries a much better choice. The NiMH battery design configuration would replace the Pb-Acid batteries already on board SAUVIM, therefore the new system was designed around the space provided. To meet the electrical requirements of the thrusters we would need to connect at least 120 of the C-cell batteries together to get the required voltage of 144V. The large 144V cell is called the M-cell. Providing enough current to the thrusters would require paralleling many M-cells together. Discharging efficiency is good within a current range of 0.1 CmA to 2CmA (1), where C is 3000mA. At the maximum discharge current of 2CmA, or 6Amps per M-cell, 25 M-cells could provide the needed 148Amps of our system. However, because our batteries will be going underwater the batteries will be in a sealed enclosure. High-current discharging can lead to heat generation and in some cases, gases (oxygen, hydrogen) may be given off, and there is a danger of the batteries bursting or

rupturing in the presence of a source of ignition (1). To prevent high-current discharging 48 M-cells were paralleled such that the maximum discharge of each cell is no greater than 3Amps, or 1CmA.

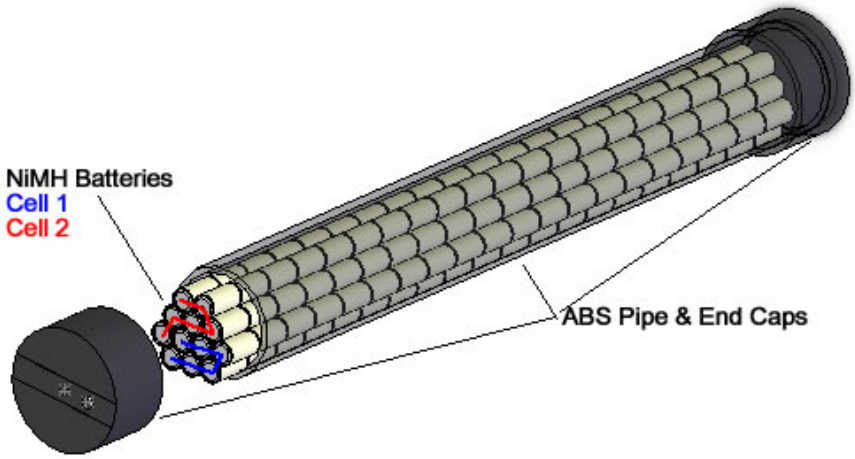


Figure 2 ABS tube with 2 M-cells

For the battery housing we used 4"ABS plastic tubes with chemically sealed end caps. Each of the ABS housings contains 2 M-cells for a total of 24 sealed enclosures. The end caps are fitted with wet pluggable 4pin electrical connectors and a pressure relief valve. The relief valve vents if the internal pressure of the ABS housing is 5psi greater than the outside ambient pressure and allows any internal gas build up to escape. Charge and discharge testing of the M-cells showed no signs of gas discharge or pressure build up inside the sealed tubes.

When we examined paralleling 48 M-cell batteries together we had to consider failure of a battery and what effects a battery failure would have on the system. Failure of a battery would include, but is not limited to, shorting due to a flooded ABS housing or other internal construction failure, an open or floating voltage resulting from a break in the series connection, and intermittent connections that occur from poor battery construction. Intermittent connection and open circuit conditions result in the loss of the battery to the system. However, shorting of one battery in parallel would mean that all the batteries would be shorted and would effectively destroy the system. Diodes in series with output of each M-cell isolate each cell from the other. Unfortunately the series diode would prevent charging unless it was placed outside the ABS housing. A junction box was introduced to the design of the battery system to simplify both discharging and charging of the batteries in a unified platform. The junction box routes power from the batteries to the thrusters and from the charger to the batteries. The charging system is external to SAUVIM and is not detailed in this report(2). Figure 3 shows a simplified circuit diagram of 2 M-cells and the power routing within the junction box. This circuit is repeated 24 times within the junction box and parallels all the batteries together yet allows each of the batteries to be independently charged. The diode configuration also prevents the high voltage of the batteries to be seen at the connection of the battery charger.

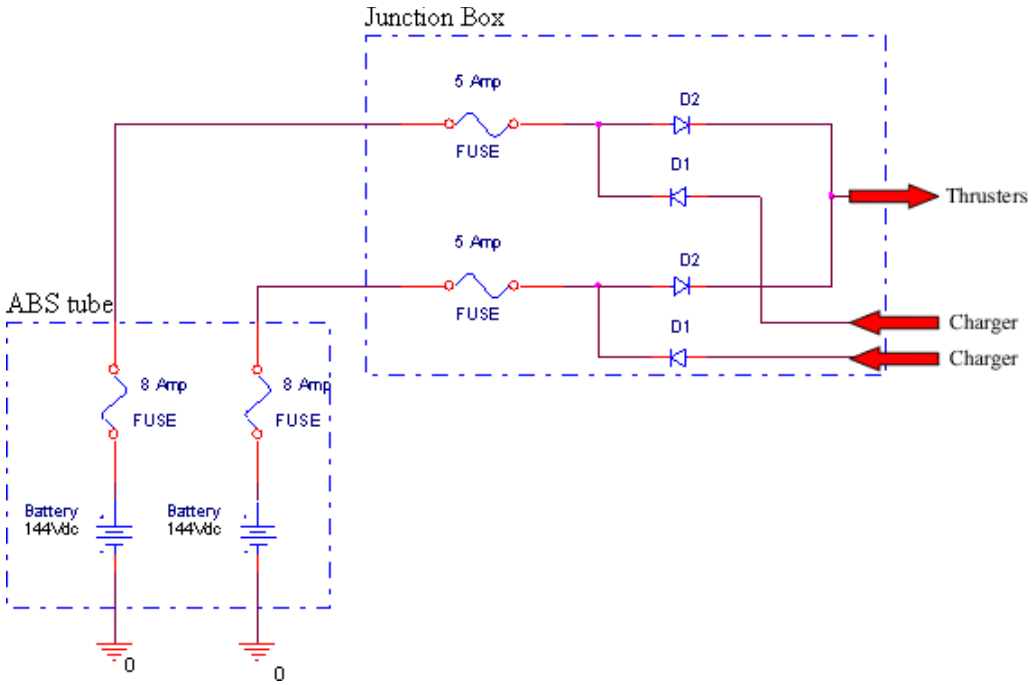


Figure 3 A simplified diagram of the power through the junction box

Over-current protection of the batteries is primarily done by the 5Amp fuse in the junction box, however in the event that a battery is shorted directly an internal 8Amp fuse is installed. In the event the 8Amp fuse needs to be replaced the ABS tube is destroyed where the 5Amp fuse can be easily replaced by opening the junction box.

The battery upgrade for the thrusters was completed in April 2007 and the performance increase was immediately apparent during initial testing. For the first time, SAUVIM's

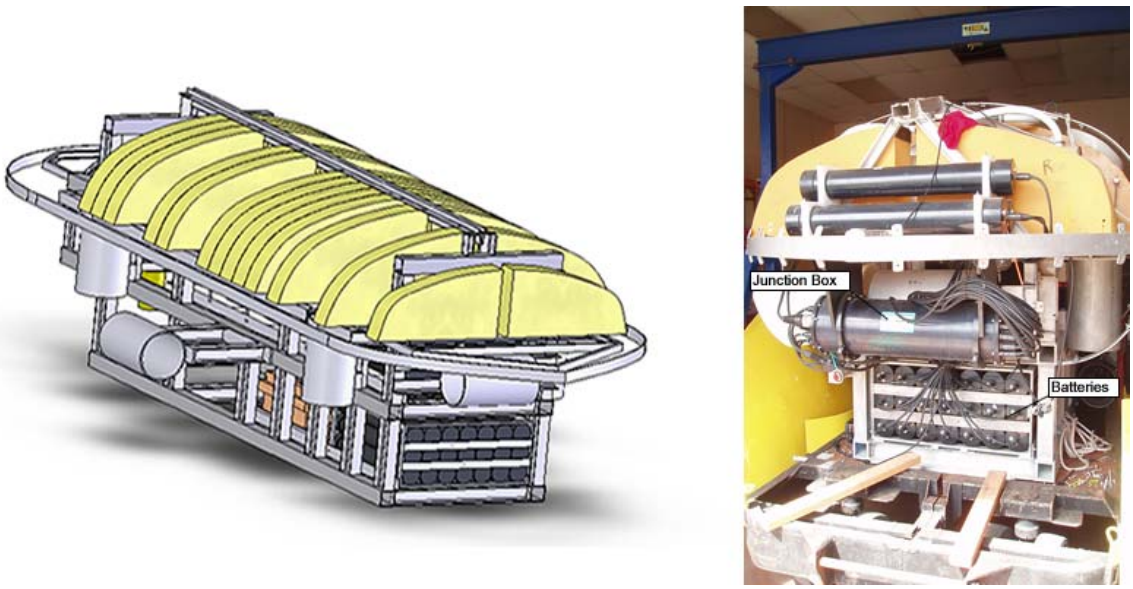


Figure 4 SAUVIM model and final installation of NiMH batteries and junction box

thrusters lasted longer than the CPU batteries. SAUVIM was tested for 6 hours a day for 3 days before needing a recharge. This is a 6 fold increase in run time over the previous system and is repeated consistently with current tests.

References

1) Nickel Metal Hydride Handbook, Panasonic, August 2005
2) High Voltage NiMH Battery Charger, MASE Inc., September 2008

References

[Aerotech92] Aerotech, “Operation & Technical Manual”, 1992.

[Akiba00] Akiba T and Kakui Yoshimi, “Design and Testing of an Underwater Microscope and Image Processing System for the Study of Zooplankton Distribution”, IEEE Journal of Oceanic Engineering, vol. 25, no 1, page 97-104 (2000)

[Aloimonos97] Aloimonos, Y., Visual Navigation: From Biological Systems to Unmanned Ground Vehicles, Mahwah, NJ, Lawrence Erlbaum Associates, 1997.

[ANSYS94] ANSYS User's Manual (Version 5.2), Ansys, Inc., Houston, TX, USA, 1994.

[ANSYS99] ANSYS User’s Manual, ANSYS, Inc., Canonsburg, PA, 1999.

[Antonelli98] Antonelli, G., and S. Chiaverini, "Task-Priority Redundancy Resolution for Underwater Vehicle-Manipulator Systems", in Proceedings of the 1998 IEEE International Conference on Robotics & Automation, pp.756-761, Leuven, Belgium, 1998.

[Archibald93] Archibald C and Petriu E, “Robot Skills Development Using a Laser Ranger Finder”, IEEE 0-7803-1229-5, page 448-452 (1993)

[Ashby80] Ashby, M. F., and D. R. H., Jones, “Engineering Materials 1: An Introduction to their Properties and Applications”, New York: Pergamon Press, 1980.

[Askeland84] Askeland, D. R., “The Science and Engineering of Materials” California: Wadsworth, Inc., 1984

[Atari97] Atari A and Dodds G, “Practical Stereo Vision and Multi-Laser Scanning in Object Face Detection and Orientation Determination”, IEEE 0-7803-4119-8, page 746-751 (1997)

[Atari99] Atari A and Dodds G, “Integration of a Stereo Multiple-laser Ranger System and Force Sensor in a Virtual Robotic Environment”, IEEE/RSJ Int. Conference on Intelligent Robots and Systems, page 1519-1524 (1999)

[Auran95] Auran, P.G. and O. Silven, “Ideas for Underwater 3D Sonar Range Sensing and Environmental Modeling”, Proceeding of CAMS’95, pp. 284-290, 1995.

[AUV96] Proceedings of the 1996 Symposium on Autonomous Underwater Vehicle Technology, Monterey, California, 1996.

[Avallone87] Avallone, E.A, and Baumeister, T., Mark's Standard Handbook for Mechanical Engineers - Ninth Edition, McGraw-Hill Book Co., 1987.

[Ayache91] Ayache, N., Artificial Vision for Mobile Robots : Stereo Vision and Multisensory Perception, MIT Press, 1991.

[Beckwith90] Beckwith, T.G., and Maranogoni, R.D., Mechanical Measurements, Addison-Wesley Publishing Company, 1990.

[Ben07] Benjamin, M. R., 2007. Software Architecture and Strategic Plans for Undersea Cooperative Cueing and Intervention. White paper, NAVSEA-DIVNPT, Code 2501

[Bollinger89] Bollinger, J.G. and N.A. Duffie, Computer Control of Machines and Processes, Reading, MA, Addison-Wesley Publishing Company, 1989.

[Borland92] Borland C++ User's Guide, Borland International, Inc., 1992.

[Brush75] Brush, D.O. and B.O. Almroth, Buckling of Bars, Plates, and Shells, New York: McGraw-Hill, Inc., 1975.

[Brutzman92] Brutzman, D.P., Y. Kanayama & M.J. Zyda, "Integrated Simulation for Rapid Development of Autonomous Underwater Vehicles," Proceedings of the IEEE Oceanic Engineering Society Autonomous Underwater Vehicle 92 Conference, Jun. 1992.

[Bushnell85] Bushnell, D., "Computerized Buckling Analysis of Shells", Dordrecht: Martinus Nijhoff Publishers, 1985

[Caimi95] Caimi, F, "Technical Challenges and Recent Developments in Underwater Imaging", Micro-Optics/Micromechanics and Laser Scanning and Shapining, M. Edward Motamedi, Leo Beiser, Editors, Proceedings of SPIE Vol. 2383, 408-418 (1995)

[Caimi99] Caimi F and Kocak D and Colquitt C, "Design and performances characterization of Simultaneous Reflectance and Surface Mapping laser Scanner for Application in Underwater inspection", in Optical Scanning: Design and Application, Leo Beiser, Stephen F. Sagan, Gerald F. Marshall, Editors, SPIE Vol. 3787, 228-239 (1999).

[Callister91] Callister, W.D., Materials Science and Engineering, John Wiley and Sons, Inc., 1991.

[Casalino00] Casalino, G., D.Angeletti, G.Cannata, G. Marani: On the Function and Algorithmic Control Architecture of the AMADEUS Dual Arm Robotic Workcell, SURT 2000, Wailea, Hawaii, June 2000.

[Chan95] Chan, T.F. and R.V. Dubey, "A Weighted Least-Norm Solution Based Scheme for Avoiding Joint Limits for Redundant Joint Manipulators", IEEE Transaction on Robotics and Automation, vol.11, no.2, pp.286-292, 1995.

[Chappell99] Chappell, S.C., R.J. Komerska, L. Peng & Y. Lu, "Cooperative AUV Development Concept (CADCON) - An Environment for High-Level Multiple AUV Simulation," Proceedings of the 11th International Symposium on Unmanned Untethered Submersible Technology, Aug. 1999.

[Chiaverini93] Chiaverini, S. and L. Sciavicco, "The Parallel Approach to Force/Position Control of Robotic Manipulators," IEEE Transaction on Robotics and Automation, vol. 9, pp. 361-373, 1993.

[Chiaverini97] Chiaverini, S., "Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators," IEEE Transaction on Robotic and Automation, vol. 13, 398-410, 1997.

[Choi95a] Choi, S.K., J. Yuh, and G.Y. Takashige, "Design of an Omni-Directional Intelligent Navigator, Underwater Robotic Vehicles: Design and Control, TSI Press, pp. 277-297, 1995.

[Choi95b] Choi, S.K. and J. Yuh, "Development of an Omni-Directional Intelligent Navigator", IEEE Robotics and Automation Magazine, 1995.

[Choi95c] Choi, S.K., G.Y. Takashige & J. Yuh, "Development of an Omni-Directional Intelligent Navigator," IEEE Robotics and Automation Magazine on Mobile Robots, Mar. 1995.

[Choi96] Choi, S.K., and J. Yuh, "Experimental Study on a Learning Control System with Bound Estimation for Underwater Robots", International Journal of Autonomous Robots, 3 (2 & 3), pp. 187-194, 1996.

[Clayton82] Clayton, B.R. and Bishop, R.E.D., Mechanics of Marine Vehicles, E. & F.N. Spon Ltd., 1982.

[Comstock67] Comstock, J.P., Principles of Naval Architecture, Society of Naval Architects and Marine Engineers, 1967.

[Coz74] Cox, A.W., Sonar and Underwater Sound, Lexington Books, 1974.

[Craig86] Craig, J.J, Introduction to Robotics: Mechanics and Control, Reading, MA, Addison-Wesley, 1986.

[Crawford98] Crawford A and Hay A, "A Simple System for Laser-Illuminated Video Imaging of Sediment Suspension and Bed Topography", IEEE Journal of Oceanic Engineering, vol. 23, no 1, page 12-19 (1998)

[Cristi96] Cristi, R., M. Caccia, G. Veruggio and A.J. Healey, "A Sonar Based Approach to AUV Localization", Proceeding of CAMS'95, pp. 291-298, 1996.

[Cunha95] Cunha, J.P., R.R. Costa, and L. Hsu, "Design of a High Performance Variable Structure Position Control of ROV's," IEEE Journal of Oceanic Engineering, vol. 20, no. 1, pp.42-55, 1995.

[D'egoulange94] D'egoulange, E. and P. Dauchez, P., "External Force Control of an Industrial PUMA 560 Robot," journal of Robotic Systems, vol. 11, pp. 523-540, 1994.

[de Wit96] Candus de Wit, C., B. Siciliano, and G. Bastin (Editors), Theory of Robot Control, Springer-Verlag, Berlin, Germany, 1996.

[DeBitetto94] DeBitetto, P.A., "Fuzzy Logic for Depth Control of Unmanned Undersea Vehicles," Proceedings of the Symposium of Autonomous Underwater Vehicle Technology, pp. 233-241, 1994.

[Deepsea90] UnderPressure Software Manual, DeepSea Power and Light Co., 1990.

[Doebelin75] Doebelin, E.O., Measurement Systems: Application and Design, New York, McGraw-Hill Book Company, 1975.

[Dote90] Dote, Y., Servo Motor and Motion Control Using Digital Signal Processors, Englewood Cliffs, NJ, Prentice-Hall Publishing, 1990.

[Dougherty90] Dougherty, F. and G. Woolweaver, "At-Sea Testing of an Unmanned Underwater Vehicle Flight Control System," Proceedings of the Symposium of Autonomous Underwater Vehicle Technology, pp. 65-73, 1990.

[Dunningan96] Dunningan, M.W., D.M. Lane, A.C. Clegg, and I. Edwards, "Hybrid Position/Force Control of a Hydraulic Underwater Manipulator," IEEE Proceedings Control Theory and Application, vol. 143, no. 2, pp. 145-151, March 1996.

[Englemann95] Engelmann, W.H., Handbook of Electric Motors, M. Dekker, 1995.

[Evans96] Evans, A.J., Basic Digital Electronics - Digital System Circuits and Their Functions, Master Publishing Inc., 1996.

[Evans03] Evans, J., Redmond, P., Plakas, C., Hamilton, K., Lane, D., 2003. Autonomous docking for Intervention-AUVs using sonar and video-based real-time 3D pose estimation. OCEANS 2003. Proceedings, 22-26 Sept., Vol. 4, pp.2201-2210.

[Ferrerri97] Ferrerri, G., G. magnani, and P. Rocco, "Toward the Implementation of Hybrid Position/Force Control in Industrial Robots," IEEE Transaction on Robotics and Automation, vol. 16, pp. 838-845, 1997.

[Fox88] Fox J, "Structured light imaging in turbid water", Underwater Imaging, Douglas J. Holloway, Editor, SPIE vol. 980, page 66-71 (1988).

[Fox92] Fox, R.W. and McDonald, A.T., Introduction to Fluid Mechanics, John Wiley amd Sons, Inc., 1992.

[Gere90] Gere, J.M. and Timoshenko, S.P., Mechanics of Materials, PWS-Kent Publishing Co., 1990.

[Geyer77] Geyer, R. A., Submersibles and Their Use in Oceanography and Ocean Engineering, Amsterdam, Elsevier Scientific Pub. Co., 1977.

[Gill85] Gill, R., Electrical Engineering Handbook, Siemens Co., 1985.

[Goheen98] Goheen, K.R., and R.E. Jeffery, "Multivariable Self-Tuning Autopilots for Autonomous and Remotely Operated underwater Vehicles", IEEE Journal of Oceanic Engineering, vol. 15, pp.144-151, 1990.

[Goldberg89] Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[Harding97] Harding, K.G. and D.J. Svetkoff (chairs/editors), Three-dimensional Imaging and Laser-based Systems for Metrology and Inspection III (Pittsburgh, PA), International Society for Optical Engineering, Bellingham, Washington, 1997.

[Healy92] Healy, A.J. and D.B. Macro, "Slow Speed Flight Control of Autonomous Underwater Vehicles: Experimental Results with NPS AUV II", Proc. of ISOPE, pp. 523-532, 1992.

[Healy93] Healy, A.J. and D. Lienard, "Multi-variable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles," IEEE Journal of Oceanic Engineering, vol. 18, no. 3, pp. 327-339, 1993.

[Hibbeler92] Hibbeler, R.C., Engineering Mechanics, Macmillian Publishing Co., 1992.

[Hill70] Hill, P.G. and Peterson, C.R., Mechanics and Thermodynamics of Propulsion, Addison Wesley Publishing Co., 1970.

[Hoerner65] Hoerner, S.F., Fluid Dynamic Drag: Practical Information on Aerodynamic and Hydrodynamic Resistance, American Institute of Aeronautics and Astronautics, 1965.

[Hollerback87] Hollerbach, J.M. and K.C. Suh, "Redundancy Resolution of Manipulator Through Torque Optimization", IEEE Journal of Robotics and Automation, vol RA-3, No.4, pp. 308-316, 1987.

[Holman89] Holman, J.P. and W.J. Gajda, Jr., Experimental Methods for Engineers, New York, McGraw-Hill Book Company, 1989.

[Howard86] Howard, G.. Automobile Aerodynamics: Theory and Practice for Road and Track, Motorbooks International, 1986.

[Hsu94] Hsu, L., R. Costa, and F. Lizarralde, "Underwater Vehicle Dynamic Positioning Based on a Passive Arm Measurement System", International Advanced Robotics Programme, pp. 23-32, 1994.

[Hudson99] Hudson, J. and Luecke J., Basic Commincations Electronics, Master Publsihing, Inc., 1999.

[Hueber95] Huebner, K. H., E. A., Thornton, and T. G., Byrom, "The Finite Element Method for Engineers," New York: John Wiley & Sons, Inc., 1995.

[Hughes94] Hughes, A., Electric Motors and Drives - Fundamentals, Types and Applications, BH Newnes, 1994.

[Hull83] Hull, D., "Axial Crusing of Fibre Reinforced Composite Tubes," Structural Crashworthiness, Eds. N. Jones and T. Wierzbicki, Butterworth., pp. 118-135, 1983.

[Hyer88] Hyer, M. W., "Respond of Thick Laminate Cylinders to External Hydrostatic Pressure," Journal of Reinforced Plastics and Composites, vol. 7, pp. 321-340, 1988.

[ICI Thermoplastic Composite92] ICI Thermoplastic Composite, "Thermoplastic Composite Handbook," 1992.

[Incropera85] Incropera, F.P. and DeWitt, D.P., Introduction to Heat Transfer, John Wiley and Sons, 1985.

[Ishii94] Ishii, K., T. Fujii, and T. Ura, "A Quick Adaptation Method in Neural Network Based Control System for AUVs," Proceedings of the Symposium of Autonomous Underwater Vehicle Technology, pp.269-274, 1994.

[Jones92] Jones, D. A., "Principle and Prevention of Corrosion," New York: Macmillian Publishing Company, 1992.

[Kajita97] Kajita, H. and K. Kosuge, "Force Control of Robot Floating on the Water Utilizing Vehicle Restoring Force," Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems, vol.1, pp. 162-167, 1997.

[Kato93] Kato, N., Y. Ito, K. Asakawa, and Y. Shirasaki, "Guidance and Control of Autonomous Underwater Vehicle AQUA Explorer 1000 for Inspection of Underwater Cables", Proc. 8th Int. Symposium on Unmanned, Untethered Submersible Technology, Sept. 1993.

[Kawaguchi96] Kawaguchi, K., C. Ikehara, S.K. Choi, M. Fujita, and J. Yuh, "Design of an Autonomous Underwater Robot: ODIN II," World Automation Congress, Montpellier, France, May 1996.

[Kernighan78] Kernighan, B.W. and D.M. Ritchie, The C Programming Language, Englewood Cliffs, NJ, Prentice-Hall, Inc., 1978.

[Klafter83] Klafter, R. D., Robotic Engineering: an Integrated Approach, Prentice Hall, 1989.

[Klein83] Klein, C.A. and C.S. Huang, "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," IEEE Trans. on System, Man, and Cybernetics, vol. SMC-13, pp. 245-250, 1983.

[Kocak99] Kocak D and Lobo N and Widder E, "Computer Vision Techniques for Quantifying, Tracking, and Identifying Bioluminescent Plankton", IEEE Journal of Oceanic Engineering, vol. 24, no 1, page 81-95 (1999)

[Kochin65] Kochin, N.E., I.A. Kibel, and N.V. Rose, Theoretical Hydrodynamics, John Wiley & Sons, 1965.

[Krar67] Krar, S.F., and Amand, J.E., Machine Shop Training, McGraw-Hill Co., 1967.

[Kuroda95] Kuroda, Y., K. Aramaki, T. Fujii & T. Ura, "A Hybrid Environment for the Development of Underwater Mechatronic Systems," Proceedings of the 1995 IEEE 21st International Conference on Industrial Electronics, Control, and Instrumentation, Nov. 1995.

[Lamb45] Lamb, H., Hydrodynamics, Dover, 1945.

[Lander87] Lander, C.W., Power Electronics, McGraw-Hill, 1987.

[Lane99] Lane D and Davies: and Robinson G and O'Brien D and Sneddon J and Seaton E and Elfstrom Anders, "The AMADEUS Dextrous Subsea Hand: Design, Modeling, and Sensor Processing", IEEE Journal of Oceanic Engineering, vol. 24, no 1, page 96-111 (1999)

[Lawry90] Lawry, M.H., I-DEAS Student Guide, Structural Dynamics Research Corp., 1990.

[Leon95] Leon, G. F. and J. C. Hall, "Case Study-Design and Testing of the Brunswick Graphite Epoxy Composite Ring-Stiffened Thermo set Cylinder," Journal of Thermoplastic Composite Materials, vol. 8., 1995.

[Lewis84] Lewis, D.J., J.M. Lipscomb, and P.G. Thompson, "The simulation of Remotely Operated Underwater Vehicle", Proceeding of ROV 1984, pp. 245-252, 1984.

[Lewis89] Lewis, E.V., Principles of Naval Architecture, Jersey City, NJ, Society of Naval Architects and Marine Engineers, 1988-1989.

[Liegeois77] Liegeois, A., "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms," IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-7, No.2, pp.868-871, 1977.

[Lines97] Lines, D., Building Power Supplies - Useful Designs for Hobbyists and Technicians, Jerry Leucke Master Publishing Inc., 1997.

[LS-DYNA99] LS-DYNA User's Manual, Livermore Software Technology Corporation, Livermore, CA, 1999.

[Luanglat97] Luanglat, C. S., and M. N Ghasemi Nejhad., "A Crash Simulation Study of Composite Materials and Structures for Electric and Hybrid Vehicles," 14th International Electric Vehicle Symposium, Proceedings, pp. 14-17, 1997.

[Lundgren99} Lundgren, J., and P. Gudmundson, " Moisture Absorption in Glass-Fiber/Epoxy Laminates with Transverse Matrix Cracks," Composites Science and Technology, vol. 59, no. 13, pp. 1983-1991, 1999.

[Mahesh91] Mahesh, H., J. Yuh, and R. Lakshmi, "A Coordinated Control of an Underwater Vehicle and Robot Manipulator", Journal of Robotic Systems, Vol.8, No.3, pp.339-370, 1991.

[Mallick93] Mallick, P. K., "Fiber-reinforced Composites: Materials, Manufacturing, and Design," New York: Marcel Dekker, Inc., 1993.

[Marani05] Marani, G., Medrano, I., Choi, S.K., Yuh, J., 2005. A client-server oriented programming language for autonomous underwater manipulation. The Proceedings of The Fifteenth (2005) International Offshore and Polar Engineering Conference, Seoul, Korea, June 19-24.

[Marco96] Marco, D.B., Autonomous Control of Underwater Vehicles and Local Area Maneuvering, Ph.D. Dissertation, Naval Postgraduate School, 1996.

[Martini84] Martini, L.J., Practical Seal Design, Marcel Dekker, Inc., 1984.

[Mattsson89] Mattsson, E., Basic Corrosion Technology for Scientists and Engineers, Ellis Horwood Ltd., 1989.

[McLain96] McLain, T.W., S.M. Rock, and M.J. Lee, "Experiments in the Coordinated Control of an Underwater Arm/Vehicle System", Autonomous Robots 3, pp. 213-232, Kluwer Academic Publisher, Netherlands, 1996.

[McMillan95] McMillan, D.O., and R. McGhee, "Efficient Dynamic Simulation of an Underwater Vehicle with a Robotic Manipulator," IEEE Trans. on Systems, Man, and Cybernetics, Vol.25, No.8, pp.1194-1206, August, 1995.

[Microsoft88a] Microsoft QuickBASIC - Learning to Use Microsoft QuickBASIC, Microsoft Corp., 1988.

[Microsoft88b] Microsoft QuickC - Learning to Use Microsoft QuickC, Microsoft Corp., 1988.

[Milne-Thomson68] Milne-Thomson, L., Theoretical Hydrodynamics, Macmillan, 1968.

[Mims98] Mims, F., Getting Started in Electronics, Radioshack Co., 1998.

[Mullen99] Mullen L and Contarino M and Laux A and Concannon: and Davis J and Strand M and Coles B, "Moduladet Laser Line Scanner for Enhanced

Underwater Imaging", Airborne and In-water Underwater Imaging, Gary D. Gilbert, Editor, SPIE vol. 3761, page 2-9 (1999).

[Nakamura85] Nakamura, Y., and H. Hanafusa, "Task Priority based Redundancy Control of Robot Manipulators", Robotics Research: The Second International Symposium, Cambridge, MA: MIT Press, pp.155-162, 1985.

[Needler85] Needler, M.A. and Baker Don E., Digital and Analog Controls, Reston Pub. Co., 1985.

[Negahdaripour90] Negahdaripour S and Yu C. H. and Shokrollahi A, "Recovering Shape and Motion From Undersea Images", IEEE Journal of Oceanic Engineering, vol 15, no 3, page 189-198 (1990)

[Nejhad91a] Ghasemi Nejhad, M. N., R. D. Cope, and S. I. Guceri, "Thermal Analysis of In-Situ Thermoplastic-Matrix Composite Filament Winding," ASME Journal of Heat Transfer, vol. 113, no. 2, pp. 304-313, 1991.

[Nejhad91b] Ghasemi Nejhad, M. N., R. D. Cope, and S. I. Guceri, "Thermal Analysis of In-Situ Thermoplastic-Matrix Composite Tape Laying," Journal of Thermoplastic Composite Materials, vol. 4, no. 1, pp. 29-45, 1991.

[Nejhad92a] Ghasemi Nejhad, M. N., J. W., Jr., Gillespie, and R. D., Cope, "Prediction of Process-Induced Stresses for In-situ Thermoplastic Filament Winding of Cylinder," Proceedings of Third International Conference CADCOMP, Computer Aided Design in Composite Material Technology, pp. 225-253, 1992.

[Nejhad92b] Ghasemi Nejhad, M. N., J. W., Jr., Gillespie, and R. D., Cope, "Processing Stresses for In-situ Thermoplastic Filament Winding Using the Divergence Method," Proceedings of ASME Winter Annual Meeting 1992, Heat Transfer Effects in Materials Processing, Guceri, S. I., and Alam M. K., Eds., HTD-vol. 233, pp. 33-43, 1992.

[Nejhad93] Ghasemi Nejhad, M. N., "Issues Related to Processability during the Manufacture of Thermoplastic Composite Using On-line Consolidation Technique," Journal of Thermoplastic Composite Materials, vol. 6, pp. 130-145, 1993.

[Nejhad94] Ghasemi Nejhad, M. N., J. W., Jr., Gillespie, and R. D., Cope, "Effects of Processing Parameter on Material Responses during In-situ Filament Winding of Thermoplastic Composites," International Journal of Materials and Product Technology, Concurrent Engineering of Advanced Materials-Integration of Mechanics and Manufacturing, vol. 9, no. 1/2/3, pp. 183-214, 1994.

[Nejhad97] Ghasemi Nejhad, M. N., "Thermal Analysis for Thermoplastic Composite Tow/Tape Preheating and Pultrusion," Journal of Thermoplastic Composite Materials, vol. 10, no. 4, pp. 504-523, 1997.

[Ng00a] Ng, R., A., Yousefpour, M., Uyema, and M. N., Ghasemi Nejhad, "Design, Analysis, Manufacture, and Test of Shallow Water Pressure Vessels using E-Glass/Epoxy Woven Composite Material for a Semi-Autonomous Underwater Vehicle, submitted to the Journal of Composite Materials, in review, 2000.

[Ng00b] Ng, R., M., Uyema, A., Yousefpour, M. N., Ghasemi Nejhad, B., Flegal, and E., Sung, "Manufacturing and Testing of Shallow Water Composite Pressure

Vessels for Semi-Autonomous Underwater Vehicle," World Automation Congress 2000 (WAC 2000), in press, June 2000.

[Nie98] Nie, J., J. Yuh, E. Kardash, and T.I. Fossen, "On-Board Sensor-Based Adaptive Control of Small UUVs in Very Shallow Water", IFAC Symposium on Control Applications for Marine Systems, 1998.

[Nie99] Nie, J., J. Yuh, E. Kardash, and T.I. Fossen, "On-Board Sensor-Based Adaptive Control of Small UUVs in Very Shallow Water", International Journal of Adaptive Control and Signal Processing, vol. 13, 1999.

[Nygards98] Nygards J and Wernersson A, "On Covariances for fusing Laser Ranger and Vison with Sensors Onboard a Moving Robot", IEEE/RSJ Int. Conference on Intelligent Robots and Systems, page 1053-1059 (1998)

[Ogata87] Ogata, K., Discrete-time Control Systems, Prentice-Hall, New Jersey, 1987.

[Omura95] Omura, G., Mastering AutoCAD 13 for Windows95, Windows3.1, and WindowsNT, Sybex, 1995.

[Parrish73] Parrish, A., Mechanical Engineer's Reference Book, Butterworths, 1973.

[Pascol93] Pascoal, A., M. J. Rendas, V. Barroso, C. Silvestre, P. Oliveria and I. Lourtie, "Simulation Study of an Integrated Guidance System for an Autonomous Underwater Vehicle", Acoustic Signal Processing for Ocean Exploration (Eds. J.M.F. Moura and I.M.G. Lourtie), pp. 587-592, 1993.

[Pickering97] Pickering, E. R., "Welding Aluminum," Journal of Advance Materials & Processing, pp. 29-30, 1997.

[Porter67] Porter, H.W., Machine Shop Operations and Setups, American Technical Society, 1967.

[Pugh70] Pugh, H., Mechanical Behavior of Materials Under Pressure, Elsevier Publishing Co., 1970.

[Raibert81] Raibert, M.H. and J.J. Craig, "Hybrid Position/Force Control of Manipulators," Transactions of the ASME Journal of Dynamic Systems, Measurement, and Control, vol. 12, pp. 126-133, 1981.

[Reynolds89] Product and Data Catalog - Reynolds Aluminum Supply Company, Reynolds Aluminum Supply Company, 1989.

[ROV91] Intervention/ROV'91 Conference & Exposition, Hollywood, Florida, Sponsored by the ROV Committee and the South Florida Section of the Marine Technology Society, 1991.

[Sagatun92] Sagatun, S.I. Modeling and Control of Underwater Vehicles: Lagrangian Approach, Dr. Ing Thesis, Norwegian Institute of Technology, 1992.

[Sayers99] Sayers, C., Remote Control Robotics, Springer, 1999.

[Scheck84] Scheck, L.A.. and Edmondson, G.C., Practical Welding, Glencoe Publishing Co., 1984.

[Schlichting79] Schlichting, H., Boundry-Layer Theory, McGraw-Hill, Inc., 1979.

[Schwartz84] Schwartz, M.M., Composite Materials Handbook, McGraw-Hill, 1984.

[Schweitzer83] Schweitzer, P.A., Corrosion and Corrosion Protection Handbook, M. Dekker, 1983.

[Serway89] Serway, R.A. and Faughn, J.S., College Physics, Saunders College Publishing, 1989.

[Shahinpour87] Shahinpoor, M., A Robot Engineering Textbook, New York, Harper & Row Publishers, 1987.

[Shames89] Shames, I.H, Introduction to Solid Mechanics, Prentice-Hall, Inc., 1989.

[Shen81] Shen, C., and G. S. Springer, “Environmental Effects in the Elastic Moduli of Composite Materials,” Environmental Effects on Composite Material, Ed. G. S. Springer, Westport: Technomic Publishing Company, Inc., 1981.

[Shigley89] Shigley, J.E. and Mischke, C.R., Mechanical Engineering Design, McGraw-Hill, Inc., 1989.

[Smith90] Smith, C.S., Design of Marine Structures In Composite Materials, Elsevier Applied Science, 1990.

[Smith96] Smith, J. and K. Sugihara, “GA toolkit on the Web”, Proc. of the First Online Workshop on Soft Computing (WSC1), pp.93-98, 1996.

[Sonmez97] Sonmez, F.O. and H.T. Hahn, "Analysis of the On-line Consolidation Process in the Thermoplastic Composite Tape Placement", Journal of Thermoplastic Composite Materials, v. 10, pp. 543-572, 1997.

[Sprong89] Sprong, M.W. and M. Vidyasagar, Robot Dynamics and Control, New York, John Wiley & Sons, 1989.

[SubTech85] Submersible Technology: Proceedings of an International Conference (Subtech '85), Aberdeen, UK, pp. 29-31, 1985.

[Sugihara97] Sugihara, K. and Yuh, J., “GA-based motion planning for underwater robotic vehicle,” Proc. 10th Int’l Symp. on Unmanned Untethered Submersible Technology (UUST-10), Durham, NH, 1997, pp.406-415.

[Sugihara98a] Sugihara, K., “GA-based on-line path planning for SAUVIM,” Proc. 11th Int’l Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA-98-AIE), Castellon, Spain, 1998, pp.329-338.

[Sugihara98b] Sugihara, K. and J. Smith, Genetic Algorithms for Adaptive Planning of Path and Trajectory of a Mobile Robot in 2D Terrain, IEICE Trans. on Information and Systems, to appear 1998 .

[Sugihara98c] Sugihara, K. and J. Yuh, “GA-based Motion Planning for Underwater Robotic Vehicles”, Proc. 10th Int’l Symp. On Unmanned Untethered Submersible Technology (UUST-10), pp.406-415, 1998.

[Sugihara99] Sugihara, K. and Smith, J., “Genetic algorithms for adaptive planning of path and trajectory of a mobile robot in 2D terrain,” IEICE Trans. Information and Systems, Vol. E82-D, No. 1, pp.309-317, January 1999.

[Svensoon99] Svensson S. and Lexander J. and Ericson B, “OBSERVATION AND INSPECTION IN SWEDISH WATERS”, Underwater Imaging, Douglas J. Holloway, Editor, SPIE vol. 980, page 75-81 (1988).

[Swartz91] Swartz: and Cummings J, “Laser range-gated underwater imaging including polarization discrimination”, Underwater Imaging, Photography, and Visibility, Richard W. Spinard, SPIE vol. 1537, page 42-56 (1991).

[Tai99] Tsai, L.W., Robot Analysis: The Mechanics of Parallel and Serial Manipulators, John Wiley and Sons, 1999.

[Takashi91] Takashi, K., Electric Motors and their Controls: An Introduction, Oxford University Press, 1991.

[Tarn96] Tarn, T.J., G.A., Shoultsand, and S.P. Yang, "A Dynamic Model of an Underwater Vehicle with a Robotic Manipulator using Kane's Method", Autonomous Robots 3, pp. 269-283, Kluwer Academic Publisher, Netherlands, 1996.

[TI92] Linear Circuits Operational Amplifiers Data Book, Texas Instruments, 1992.

[Tsai99] Tsai, L.W., Robot Analysis - The Mechanics of Parallel and Serial Manipulators, John Wiley and Sons, Inc. 1999.

[Tupper96] Tupper, E.C., Introduction to Naval Architecture, Oxford, Butterworth-Heinemann Publishing, 1996.

[Ullman92] Ullman, D.G., The Mechanical Design Process, McGraw-Hill Inc., 1992.

[Unimate81] Unimate PUMA Robot: Volume 1 – Technical Manual 398H1, Unimation Inc., Condec Company, Danbury CT, 1981.

[Unimate84] Unimate PUMA Mark II Robot: 500 Series, Volume 1 – Equipment Manual, Unimation, Westinghouse Corporation, Danbury CT, 1984.

[Unimate86] Unimate Industrial Robot: Programming Manual, User's Guide to VAL II Version 2.0 (398AG1), Unimation, Westinghouse Corporation, Danbury CT, 1986.

[Unimate97] Unimate – Supplement to the User's Guide to VAL II: VAL II-IVM PC Supervisor Interface (397W1), Unimation, Westinghouse Corporation, Danbury CT, 1987.

[Valentine98] Valentine, R., Motor Control Electronics Handbook, McGraw-Hill, 1998.

[Vieville97] Vieville, T., A Few Steps Towards Active 3D Vision, Springer-Verlag, 1997.

[Vinson87] Vinson, J.R., and R.L. Sierakowski, The Behavior of Structures Composed of Composite Materials, Netherlands, Kluwer Academic Publishers, 1987.

[Wang95] Wang, H.H., Rock, S.M., Lees, M.J., 1995. Experiments in automatic retrieval of underwater objects with an AUV. OCEANS '95. MTS/IEEE. Challenges of Our Changing Global Environment. Conference Proceedings, vol.1, pp.366-373, October 9-12.

[Weast81] Weast, R.C. and Astle, M.J., CRC Handbook of Chemistry and Physics - 61st Edition, CRC Press, Inc. 1981.

[Wedermann89] Werdermann, C., K. Friedrich, M. Cirino, and R. B. Pipes, "Design and Fabrication an On-Line Consolidation Facility for Thermoplastic Composites," Journal of Thermoplastic Composite Materials, vol. 2, pp. 293-306, 1989.

[Wells91] Wells W, "Indirect illumination to reduce veiling luminance in seawater", Underwater Imaging, Photography, and Visibility, Richard W. Spinard, SPIE vol. 1537, page 2 (1991).

[Werdermann89] Werdermann, C., K. Friedrich, M. Cirino, and R.B. Pipes, "Design and Fabrication an On-line Consolidation Facility for Thermoplastic Composites", Journal of Thermoplastic Composite Materials, vol. 2, pp. 293-306, 1989.

[Whitney87] Whitney, D.E., "Historical Perspective and State of the Art in Robot Force Control," International Journal of Robotic Research, vol. 6, no.1, pp. 3-14, 1987.

[Wilson92] Wilson, P.A., International Conference on Manoeuvring and Control of Marine Craft, Proceedings of the Second International Conference, 1992.

[Wit98] Wit, C.C.D., E.O. Diaz, and M. Perrier, "Robust Nonlinear Control of an Underwater Vehicle/ Manipulator System with Composite Dynamics", Proc. IEEE Conf. on Robotics and Automation, pp.452-457, 1998.

[Yang98a] Yang, K.C., J. Yuh, and S.K. Choi, "Experimental Study of Fault-tolerant System Design for Underwater Robots", Proc. IEEE Conf. on Robotics and Automation, pp. 1051-1056, 1998a.

[Yang98b] Yang, K.C., J. Yuh, and S.K. Choi, "Experimental Study of Fault-tolerant System Design for Underwater Robots", Journal of System Sciences, 1998b.

[Yousefpour00a] Yousefpour, A., and M. N. Ghasemi Nejhad, "Effects of Geometric Optimization of Tapered End-caps on Thick Thermoplastic Composite Pressure Vessels for Deep Ocean Applications," Journal of Thermoplastic Composite Materials, submitted for publication, 2000.

[Yousefpour00b] Yousefpour, A., and M. N. Ghasemi Nejhad, "Experimental and Computational Study of APC-2/AS4 Thermoplastic Composite C-Rings," Journal of Thermoplastic Composite Materials, in press, 2000b.

[Yousefpour00c] Yousefpour, A., R., Ng, M., Uyema, M. N., and Ghasemi Nejhad, "Design and Finite Element Analysis of Shallow Water Composite Pressure Vessels for Semi-Autonomous Underwater Vehicle," World Automation Congress 2000 (WAC 2000), in press, June 2000c.

[Yousefpour99] Yousefpour, A., and M. N. Ghasemi Nejhad, "Testing and Finite Element Modeling of APC-2/AS4 Thermoplastic Composite C-rings," 31st International SAMPE Technical Conference: Advanced Materials & Processes Preparing for the New Millennium, vol. 31, pp. 643-654, Chicago, IL, 1999.

[Yuh92] Yuh, J., V. Adivi & S.K. Choi, "Development of a 3D Graphic Test Platform for Underwater Robotic Vehicles," Proceedings of the 2nd International Offshore and Polar Engineering Conference, Jun. 1992.

[Yuh94a] Yuh, J., "Learning Control for Underwater Robotic Vehicles", IEEE Control System Magazine, vol.15, No.2, pp.39-46, 1994.

[Yuh94b] Yuh, J., Underwater Robotic Vehicles: Design and Control, Workshop on Future Research Directions in Underwater Robotics, TSI Press, p. 361, 1994.

[Yuh95] Yuh, J., Underwater Robotic Vehicles: Design and Control, TSI Press, 1995.

[Yuh96] Yuh, J., "An Adaptive and Learning Control System for Underwater Robots", 13th World Congress International Federation of Automatic Control, A, pp. 145-150, 1996.

[Yuh98a] Yuh, J., S.K. Choi, C. Ikehara, G.H. Kim, G. McMurtry, M. Ghasemi Nejhad, N. Sarkar, and K. Sugihara, "Design of a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM)," Proceeding of the Underwater Technology '98, 1998.

[Yuh98b] Yuh, J., J. Nie, and W.C. Lee, "Adaptive Control of Robot Manipulators Using Bound Estimation", IEEE International Conference on Intelligent Robots and Systems, 1998.

[Yuh99] Yuh, J. and J. Nie, "Experimental Study on Adaptive Control of Underwater Robots," Proceedings of the IEEE International Conference of Robotics and Automation, Detroit, MI, May 1999.

[Zege91] Zege E and Ivanov A and Katsev I, "Image Transfer Through a Scattering Medium" Springer-Verlag, Berlin, page 277-305 (1991)

[Zuech88] Zuech, N., Applying Machine Vision, John Wiley and Sons, Inc., 1988.

Publications

Choi, S.K., Easterday, O., Ikehara, C., Coutsourakis, C., & Yuh, J., "Development of SAUVIM," The Proceedings of the *Symposium of Underwater Robotic Technologies* (*SURT*), Maui, HI, Jun. 2000.

Choi, S.K., Sugihara, K., Menor, S., Nip, A., & Yang, Z., "A Predictive Virtual Environment Monitor for SAUVIM," The Proceedings of the *Symposium of Underwater Robotic Technologies* (*SURT*), Maui, HI, Jun. 2000.

Choi, S.K., Menor, S. & Yuh, J., "Distributed Virtual Environment Collaborative Simulator for Underwater Robots" The Proceedings of the *International Conference on Intelligent Robots and Systems* (*IROS* '00), Takamatsu, JAPAN, Nov. 2000

Choi, S.K. & Easterday, O.T., "The Development of an Underwater Vehicle Monitoring System and Its Sensor Systems," The Proceedings of the *International Symposium on Experimental Robotics* (*ISER*), Honolulu, HI, Dec. 2000.

Choi, S.K. and Yuh, J., "A Virtual Collaborative World Simulator for Underwater Robots using Multi-Dimensional, Synthetic Environment," The Proceedings of the *IEEE International Conference on Robotics and Automation* (*ICRA* '01), Seoul, KOREA, May 2001. dfsdfs

Hanai, A., McLeod, C., Rosa, K., Marani, G., Choi, S. K., A Practical Approach to the Development of Thruster Models for Underwater Robots, The 18th International Offshore (Ocean) and Polar Engineering Conference & Exhibition, Vancouver, British Columbia, Canada, July 6-11, 2008

Kim, J., G. Marani, W. K. Chung, J. Yuh, "Kinematic Singularity Avoidance for Autonomous Manipulation in Underwater," The Fifth *ISOPE Pacific/Asia Offshore Mechanics Symposium*, Daejeon, Korea, November 17-20, 2002.

Kim, J., G. Marani, W. K. Chung, J. Yuh: *Task Reconstruction Method for Real-Time Singularity Avoidance for Robotic Manipulators*, Advanced Robotics, Vol. 20, No. 4, pp. 391-498, 2006

Kim, T.W., Choi, S.K., Lee, J.W., West, M.E., & Yuh, J., "A Real-Time Distributed Control Architecture for AUVs," The Proceedings of the *Symposium of Underwater Robotic Technologies* (*SURT*), Maui, HI, Jun. 2000.

Kim, T.W. and J. Yuh, "A Novel Neuro-Fuzzy Controller for Autonomous Underwater Vehicles, " Proc. of *IEEE International Conference on Robotics and Automation (ICRA 2001)*, pp.2350-2355, May 2001 (Seoul, Korea).

Kim, T.W. and J. Yuh, "Task Description Language for Underwater Robots," Proc. of *IEEE International Conference on Intelligent Robots and Systems* (*IROS* '03), Vol.1, pp.565-570, Oct. 2003 (Las Vegas, NV, USA).

Kim, T.W. and J. Yuh, "Application of On-line Neuro-Fuzzy Controller to AUVs," *Journal of Information Sciences*, Vol.145, No.1, pp.169-182, Aug. 2002

Kim, T.W., J. Yuh, and S. Choi, "Real-Time Control Architecture," *IFAC Workshop on Guidance and Control of Underwater Vehicle* (*GCUV* '03), South Wales, UK, April 2003

Kim, T.W. and J. Yuh, "Development of real-time control architecture for a semi-autonomous underwater vehicle for intervention missions," *Journal of Control Engineering Practice*, Vol.12/12, pp.1521-1530, Sep. 2004

Marani, G., Bozzo, T., & Choi, S.K., "A Fast Prototyping Approach for Designing the Maris Manipulator Control," The Proceedings of the *Symposium of Underwater Robotic Technologies* (*SURT*), Maui, HI, Jun. 2000.

Marani, G., "Development of a Dexterous Redundant Manipulator for Automation of Machine Tools," 2002 International Intensive Short Course on Machine Tools and new Machining Technology, Changwon, Korea, Nov. 7-9, 2002.

Marani, G., J. Kim, J. Yuh, W. K. Chung, "Algorithmic singularities avoidance in task-priority based controller for redundant manipulators," *IEEE/RSJ International Conference on Intelligent Robots and Systems* (*IROS* '03), Las Vegas, October 27-31, 2003

Marani, G., J. Yuh, S. K. Choi, "Autonomous manipulation for an intervention AUV," in *Guidance and Control of Unmanned Marine Vehicles* (G. Roberts and B. Sutton, Eds.), IEE's Control Engineering Series, 2004.

Marani, G, I. Medrano, S. K. Choi, J. Yuh: *A client-server oriented programming language for autonomous underwater manipulation*, The Proceedings of The Fifteenth (2005) International Offshore and Polar Engineering Conference, Seoul, Korea, June 19-24, 2005.

Marani, G., Choi, S.K., Yuh, J.: *Experimental Study on Autonomous Manipulation for Underwater Intervention Vehicles*, ISOPE 2007: Proceedings of the 17th International Offshore and Polar engineering Conference, Libon, Portugal, July 1-6 2007.

Ng, R., Uyema, M., Yousefpour, A., Ghasemi Nejhad, M.N., Flegal, B., and Sung, E., "Manufacturing and Testing of Shallow Water Composite Pressure Vessels for Semi-Autonomous Underwater Vehicle," *World Automation Congress* (WAC 2000), June 2000.

Sugihara, K. and Yuh, J., "GA-based motion planning for underwater robotic vehicle," Proc. 10th *Int'l Symp. on Unmanned Untethered Submersible Technology* (*UUST*-10), Durham, NH, 1997, pp.406-415.

Sugihara, K., "GA-based on-line path planning for SAUVIM," Proc. 11th Int'l Conf. on *Industrial and Engineering Applications of Artificial Intelligence and Expert Systems* (IEA-98-AIE), Castellon, Spain, 1998, pp.329-338.

Sugihara, K. and Smith, J., "Genetic algorithms for adaptive planning of path and trajectory of a mobile robot in 2D terrain," *IEICE Trans. Information and Systems*, Vol. E82-D, No. 1, pp.309-317, January 1999.

Sugihara, K. and Yuh, J., "Adaptive, intelligent motion planning system for AUV's," The Proceedings of the *Symposium of Underwater Robotic Technologies* (*SURT*), Maui, HI, Jun. 2000.

Yousefpour, A., and Ghasemi Nejhad, M. N., "Testing and Finite Element Modeling of APC-2/AS4 Thermoplastic Composite C-rings," 31st *International SAMPE* (*Society for the Advancement of Material and Process Engineering*) *Technical Conference*, vol. 31, pp. 643-654, Chicago, IL, October 1999.

Yousefpour, A., Ng, R., Uyema, M., and Ghasemi Nejhad, M.N., "Design and Finite Element Analysis of Shallow Water Composite Pressure Vessels for Semi-Autonomous Underwater Vehicle," *World Automation Congress* 2000 (*WAC* 2000), June 2000.

Yousefpour, A. and M. N., Ghasemi Nejhad "Experimental and Computational Study of APC-2/AS4 Thermoplastic Composite C-Rings," *Journal of Thermoplastic Composite Materials*, 2000.

Yousefpour, A., and M. N. Ghasemi Nejhad, "Effects of Geometric Optimization of Tapered End-caps on Thick Thermoplastic Composite Pressure Vessels for Deep Ocean Applications," *Journal of Thermoplastic Composite Materials*, 2000.

Yu, Son-Cheol; Tae-Won Kim; Marani, G.; Choi, S.K., "Real-Time 3D Sonar Image Recognition for Underwater Vehicles," *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, 2007. Symposium on* , vol., no., pp.142-146, 17-20 April 2007

Yu, Son-Cheol, Tae Won Kim, G. Marani, S. K. Choi, "*High resolution acoustic camera model based object recognition for AUVs*", International Symposium on Unmanned Untethered Submersible Technology, Aug. 2007 (Durham, NH)

Yuh, J, Choi, S.K., Ikehara, C., Kim, G.H., McMurtry, G., Nejhad, M., Sarkar, N., & Sugihara, K., "Design of a Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM)," The Proceedings of the *IEEE Oceanic Engineering Society Underwater Technology* '98, Tokyo, JAPAN, Apr. 1998.

Yuh, J. and Choi, S.K., "Semi-Autonomous Underwater Vehicle for Intervention Missions (SAUVIM)," *Sea Technology*, Oct. 1999.

Yuh, J., S. Zhao, "Application of Adaptive Disturbance Observer Control to an Underwater Manipulator," *IEEE International Conference on Robotics and Automation* (*ICRA* '01), Seoul, May 21-26, 2001

Zhao, S., J. Yuh, "Experimental Study on Advanced Underwater Robot Control," *IEEE Transactions on Robotics*, Vol. 21, No. 4, August 2005.

Zhao, S., J. Yuh, S. Choi, "Adaptive DOB Control for AUVs: Experiment," *IEEE International Conference on Robotics and Automation* (*ICRA* '04), New Orleans, LA, USA, April 26-30 2004

Zhao, S., J. Yuh, "Adaptive DOB Control of Underwater Robots," *IEEE International Conference on Intelligent Robots and Systems* (*IROS* '03), Las Vegas, Nevada, October 27-31, 2003

Zhao, S., J. Yuh, "Adaptive DOB Control of Underwater Robotic Vehicles," *IEEE/MTS Oceans Conference*, 2001, Honolulu, Hawaii, Nov. 5-8, 2001